On the Security of Two RFID Mutual Authentication Protocols

Seyed Farhad Aghili[1], Nasour Bagheri[1(✉)], Praveen Gauravaram[2], Masoumeh Safkhani[3], and Somitra Kumar Sanadhya[4]

[1] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran
NBagheri@srttu.edu

[2] Innovation Labs Hyderabad, Tata Consultancy Services Limited, Hyderabad, India
P.Gauravaram@tcs.com

[3] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran
M_Safkhani@iust.ac.ir

[4] Indraprastha Institute of Information Technology, Delhi, India
Somitra@iiitd.ac.in

Abstract. In this paper, the security of two recent RFID mutual authentication protocols are investigated. The first protocol is a scheme proposed by Huang et al. [7] and the second one by Huang, Lin and Li [6]. We show that these two protocols have several weaknesses. In Huang et al.'s scheme, an adversary can determine the 32-bit secret *Access* password with a probability of 2^{-2}, and in Huang-Lin-Li scheme, a passive adversary can recognize a target tag with a success probability of $1-2^{-4}$ and an active adversary can determine all 32 bits of *Access* password with success probability of 2^{-4}. The computational complexity of these attacks is negligible.

Keywords: RFID · EPC Class-1 Generation-2 · PadGen function

1 Introduction

Radio frequency identification (RFID) uses radio frequency signals to identify objects or people automatically. Typically, the main components of an RFID system are an RFID tag, RFID reader and a back-end server [14]. The main function of an RFID system is identification and authentication. Hence most of the RFID applications need to provide authentication between a tag and a reader. Authentication is a process in which one party is assured of the identity of the another party by obtaining the required evidences, which is done in a corroborative manner. In our case these parties are the Tag and Reader/back-end database. A secure authentication protocol is expected to resist against the attacks in the scenarios such as rogue scanning, replay attack and tag counterfeiting or cloning.

On the other hand, several interconnected standards exist for RFID systems. Among them, ISO [8] and Electronic Product Code (EPC) global [5] have played

M. Hutter and J.-M. Schmidt (Eds.): RFIDsec 2013, LNCS 8262, pp. 86–99, 2013.
DOI: 10.1007/978-3-642-41332-2_6,

the main role. The EPC Class-1 Generation-2 (C1 G2) is a universal standard for low-cost passive RFID tags. This group of tags is also covered by ISO 18000-6C standard.

EPC-C1 G2 specifies that any RFID tag compliant with this standard should contain two 32-bit passwords denoted by the access password and the kill password respectively. The access password is used to authenticate the reader that wish to access information inside the tag and control access to the information. The kill password is generally used to disable the tag. A killed tag is rendered in silence thereafter and does not respond to any query from any reader. EPC-C1 G2 standard proposes a simple authentication protocol that allows a tag to authenticate a reader. This protocol attempts to protect the access password by using a simple form of masking before transmission over a wireless channel. This masking which is known as pad generation (*PadGen*) is a simple bitwise XOR. However, a passive adversary monitoring the exchanged messages between the reader and the tag can retrieve this sensitive information easily [1,13]. These results have motivated researchers to try to propose EPC-compliant authentication protocols to improve its security level. However, the main difficulty in providing a mutual authentication protocol for RFID systems with passive tags is the very limited storage and computational capabilities of EPC- C1 G2 tags that significantly limits their support for conventional cryptographic primitives such as AES. To provide the desired security of the tags that support this standard, several mutual authentication protocols [2–4,11] were proposed. In this direction, Konidala et al. have proposed an RFID mutual authentication protocol to solve ISO 18000-6C protocol weaknesses [9]. However, the designed protocol is known to be flawed and the adversary can retrieve most of the secret passwords' bits efficiently [12]. To solve Konidala et al. protocol's weakness two novel protocols described in [6,7] have recently been proposed. In this paper, these protocols are denoted by HYCLT and HLL respectively. These protocols do not use any standard cryptographic primitives and attempt to provide the desired security by simple logical operations. Note that Ma et al. [10] have shown that any RFID protocol without using PRF is subject to some kind of tag tracing attacks. We show that this is indeed the case for the current protocols both of which do not utilize a PRF. We investigate the security level of these protocols and present practical attacks to retrieve tag's secret parameters.

Our contribution: Any tag in HYCLT and HLL protocols have two 32-bit passwords called *Kill* password and *Access* password respectively. We investigate the security of protocols against secret disclosure attack and show that an adversary can determine whole *Access* password of HYCLT with a probability of 2^{-2} at a cost of a single query to the target tag. We also analyze the security of HLL protocol and present several tag recognizing attacks against it. In the presented attacks, given a tag, the adversary can recognize whether it is the target tag with the probability of $1 - 2^{-4}$. In addition, we show that a man in the middle adversary can determine whole *Access* and *Kill* passwords with success probability of 2^{-4} for negligible complexity.

Paper organization: The rest of the paper is organized as follows: In Sect. 2 we present HYCLT protocol description and discuss its security. In Sect. 3 we describe the HLL protocol and investigate its security. In Sect. 4 we conclude the paper.

2 HYCLT Mutual Authentication Protocol

Konidala et al. [9] have proposed an RFID mutual authentication protocol to solve ISO 18000-6C protocol weaknesses [9] by using a special PadGen function to mask tag's *Access* password $Apwd = Apwd_L \| Apwd_M$ before the data is transmitted. However, Konidala et al. protocol suffers from correlation attack [12]. To solve Konidala et al. protocol's weakness, Huang et al. have proposed an improved version based on a different PadGen and also successfully demonstrated the FPGA hardware implementation of their proposed mutual authentication protocol [7]. We denote this protocol by HYCLT. The notation used in the paper are depicted in Table 1.

The PadGen function proposed in HYCLT accepts a 32-bit value and two 16-bit values as input and outputs 16 bits. Given $\mathcal{X} \in \{0,1\}^{32}$, we can represent it as $\mathcal{X} = \mathcal{X}|_0\mathcal{X}|_1 \ldots \mathcal{X}|_{31}$, where $\mathcal{X}|_i \in \{0,1\}$, and given 16-bit values $\mathcal{Y} \in \{0,1\}^{16}$ and $\mathcal{Z} \in \{0,1\}^{16}$, they can be represented as $\mathcal{Y} = d_{\mathcal{Y}1}d_{\mathcal{Y}2}d_{\mathcal{Y}3}d_{\mathcal{Y}4}$ and $\mathcal{Z} = d_{\mathcal{Z}1}d_{\mathcal{Z}2}d_{\mathcal{Z}3}d_{\mathcal{Z}4}$, where $d_{\mathcal{Z}i} \in \{0,1,\ldots,15\}$, $i \in \{1,2,3,4\}$ and used as base 10(decimal) representation of a four-bit binary string. For example, $\mathcal{Z} =$ 1101 0110 1000 1001 can be represented as $\mathcal{Z} = 13\ 06\ 08\ 09$ which means that $d_{\mathcal{Z}1} = 13, d_{\mathcal{Z}2} = 06, d_{\mathcal{Z}3} = 08, d_{\mathcal{Z}4} = 09$. Similarly, one can represent $\mathcal{Y}$ and $\mathcal{Z}$ as $\mathcal{Y} = h_{\mathcal{Y}1}h_{\mathcal{Y}2}h_{\mathcal{Y}3}h_{\mathcal{Y}4}$ and $\mathcal{Z} = h_{\mathcal{Z}1}h_{\mathcal{Z}2}h_{\mathcal{Z}3}h_{\mathcal{Z}4}$, where $h_{\mathcal{Z}i} \in \{0,1,\ldots,F\}$, $i \in \{1,2,3,4\}$ and used as base hexadecimal (base 16) representation of a four-bit binary string. For example, $\mathcal{Z} =$ 1101 0110 1000 1001 can be represented as

Table 1. Notation

Notation	Description
R_i	RFID reader i
T_i	RFID tag i
Req_R	Reader request
R_{Tx}	Random numbers generated by the tag
R_{Mx}	Random numbers generated by the server
EPC	Electronic product code
$Apwd$	*Access* password
$Apwd_L$	16 least significant bits of $Apwd$
$Apwd_M$	16 most significant bits of $Apwd$
$Kpwd$	*Kill* password
$X\|_i$	ith bit of string X
$\mathcal{X}\|_{m\sim n}$	A fraction of $\mathcal{X}$ from the mth bit to the nth bit
d_{Xi}	Decimal equivalent(base 10) of the ith 4-bit of string X
h_{Xi}	Hexadecimal(base 16) equivalent of the ith 4-bit of string X
$\oplus$	Exclusive or operation
$\|$	Concatenation operation

$\mathcal{Z} = D\ 6\ 8\ 9$ which means that $h_{\mathcal{Z}1} = D, h_{\mathcal{Z}2} = 6, h_{\mathcal{Z}3} = 8, h_{\mathcal{Z}4} = 9$. Given these definitions $PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is calculated as follows:

$$\begin{aligned} PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = {} & \mathcal{X}|_{d_{\mathcal{Y}1}}\mathcal{X}|_{d_{\mathcal{Y}1}+16}\mathcal{X}|_{d_{\mathcal{Y}2}}\mathcal{X}|_{d_{\mathcal{Y}2}+16}\|\mathcal{X}|_{d_{\mathcal{Z}1}}\mathcal{X}|_{d_{\mathcal{Z}1}+16}\mathcal{X}|_{d_{\mathcal{Z}2}}\mathcal{X}|_{d_{\mathcal{Z}2}+16}\| \\ & \mathcal{X}|_{d_{\mathcal{Y}3}}\mathcal{X}|_{d_{\mathcal{Y}3}+16}\mathcal{X}|_{d_{\mathcal{Y}4}}\mathcal{X}|_{d_{\mathcal{Y}4}+16}\|\mathcal{X}|_{d_{\mathcal{Z}3}}\mathcal{X}|_{d_{\mathcal{Z}3}+16}\mathcal{X}|_{d_{\mathcal{Z}4}}\mathcal{X}|_{d_{\mathcal{Z}4}+16} \end{aligned}$$

For example assume that:

$$\begin{aligned} \mathcal{X} &= 1001\ 1111\ 0011\ 1011\ 0000\ 0011\ 1100\ 0101 \\ \mathcal{Y} &= 0111\ 0100\ 0110\ 1011 \\ \mathcal{Z} &= 1101\ 0110\ 1000\ 1001 \end{aligned}$$

then $PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is calculated as follows:

$$\begin{aligned} PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = {} & \mathcal{X}|_7\mathcal{X}|_{7+16}\mathcal{X}|_4\mathcal{X}|_{4+16}\|\mathcal{X}|_{13}\mathcal{X}|_{13+16}\mathcal{X}|_6\mathcal{X}|_{6+16}\| \\ & \mathcal{X}|_6\mathcal{X}|_{6+16}\mathcal{X}|_{11}\mathcal{X}|_{11+16}\|\mathcal{X}|_8\mathcal{X}|_{8+16}\mathcal{X}|_9\mathcal{X}|_{9+16} \\ = {} & 1110\ 0111\ 1110\ 0101 \end{aligned}$$

where:

$$\begin{aligned} \mathcal{X} = 1001\ & \underbrace{1}_{\mathcal{X}|_4}\ 1\ \underbrace{1}_{\mathcal{X}|_6}\ \underbrace{1}_{\mathcal{X}|_7}\ \ \underbrace{0}_{\mathcal{X}|_8}\ \underbrace{0}_{\mathcal{X}|_9}\ 1\ \underbrace{1}_{\mathcal{X}|_{11}}\ 1\ \underbrace{0}_{\mathcal{X}|_{13}} \\ 11\ 0000\ & \underbrace{0}_{\mathcal{X}|_{20}}\ 0\ \underbrace{1}_{\mathcal{X}|_{22}}\ \underbrace{1}_{\mathcal{X}|_{23}}\ \ \underbrace{1}_{\mathcal{X}|_{24}}\ \underbrace{1}_{\mathcal{X}|_{25}}\ 0\ \underbrace{0}_{\mathcal{X}|_{27}}\ 0\ \underbrace{1}_{\mathcal{X}|_{29}}\ 01 \end{aligned}$$

In HYCLT protocol, the tag and the server use the PadGen function to generate four masking values denoted by $PAD_1, PAD_2, PAD_3,$ and, PAD_4 respectively. Let us to represent the 32-bit *Access* password *Apwd* and the 32-bit *Kill* password *Kpwd* as $Apwd = a|_0a|_1a|_2a|_3 \ldots a|_{31}$ and $Kpwd = k|_0k|_1k|_2k|_3 \ldots k|_{31}$ respectively where $a|_i \in \{0,1\}$ and $k|_i \in \{0,1\}$. Given 16-bit random numbers R_{Tx} and R_{Mx}, for $x \in \{1,2,3,4\}$, they can be represented as $R_{Tx} = d_{R_{Tx}1}d_{R_{Tx}2}d_{R_{Tx}3}d_{R_{Tx}4}$ and $R_{Mx} = d_{R_{Mx}1}d_{R_{Mx}2}d_{R_{Mx}3}d_{R_{Mx}4}$.

The PadGen function of HYCLT protocol is used to compute masking values PAD_x, for $x \in \{1,2,3,4\}$, as follows:

$$\begin{aligned} R_{Vx} &= PadGen(APwd, R_{Tx}, R_{Mx}) \\ &= a|_{d_{R_{Tx}1}}a|_{d_{R_{Tx}1}+16}a|_{d_{R_{Tx}2}}a|_{d_{R_{Tx}2}+16}\|a|_{d_{R_{Mx}1}}a|_{d_{R_{Mx}1}+16}a|_{d_{R_{Mx}2}}a|_{d_{R_{Mx}2}+16}\| \\ &\quad a|_{d_{R_{Tx}3}}a|_{d_{R_{Tx}3}+16}a|_{d_{R_{Tx}4}}a|_{d_{R_{Tx}4}+16}\|a|_{d_{R_{Mx}3}}a|_{d_{R_{Mx}3}+16}a|_{d_{R_{Mx}4}}a|_{d_{R_{Mx}4}+16} \\ &= d_{R_{Vx}1}d_{R_{Vx}2}d_{R_{Vx}3}d_{R_{Vx}4} \end{aligned}$$

and

$$\begin{aligned} PAD_x &= PadGen(Kpwd, R_{Vx}, R_{Tx}) \\ &= k|_{d_{R_{Vx}1}}k|_{d_{R_{Vx}1}+16}k|_{d_{R_{Vx}2}}k|_{d_{R_{Vx}2}+16}\|k|_{d_{R_{Tx}1}}k|_{d_{R_{Tx}1}+16}k|_{d_{R_{Tx}2}}k|_{d_{R_{Tx}2}+16}\| \\ &\quad k|_{d_{R_{Vx}3}}k|_{d_{R_{Vx}3}+16}k|_{d_{R_{Vx}4}}k|_{d_{R_{Vx}4}+16}\|k|_{d_{R_{Tx}3}}k|_{d_{R_{Tx}3}+16}k|_{d_{R_{Tx}4}}k|_{d_{R_{Tx}4}+16} \\ &= h_{PAD_x1}h_{PAD_x2}h_{PAD_x3}h_{PAD_x4} \end{aligned}$$

where R_{Vx} is a temporary variable. For example, PAD_1 is calculated as follows:

$$\begin{aligned} R_{V1} &= PadGen(APwd, R_{T1}, R_{M1}) \\ &= a|_{d_{R_{T1}1}} a|_{d_{R_{T1}1}+16} a|_{d_{R_{T1}2}} a|_{d_{R_{T1}2}+16} \| a|_{d_{R_{M1}1}} a|_{d_{R_{M1}1}+16} a|_{d_{R_{M1}2}} a|_{d_{R_{M1}2}+16} \| \\ &\quad a|_{d_{R_{T1}3}} a|_{d_{R_{T1}3}+16} a|_{d_{R_{T1}4}} a|_{d_{R_{T1}4}+16} \| a|_{d_{R_{M1}3}} a|_{d_{R_{M1}3}+16} a|_{d_{R_{M1}4}} a|_{d_{R_{M1}4}+16} \\ &= d_{R_{V1}1} d_{R_{V1}2} d_{R_{V1}3} d_{R_{V1}4} \end{aligned}$$

and

$$\begin{aligned} PAD_1 &= PadGen(Kpwd, R_{V1}, R_{T1}) \\ &= k|_{d_{R_{V1}1}} k|_{d_{R_{V1}1}+16} k|_{d_{R_{V1}2}} k|_{d_{R_{V1}2}+16} \| k|_{d_{R_{T1}1}} k|_{d_{R_{T1}1}+16} k|_{d_{R_{T1}2}} k|_{d_{R_{T1}2}+16} \| \\ &\quad k|_{d_{R_{V1}3}} k|_{d_{R_{V1}3}+16} k|_{d_{R_{V1}4}} k|_{d_{R_{V1}4}+16} \| k|_{d_{R_{T1}3}} k|_{d_{R_{T1}3}+16} k|_{d_{R_{T1}4}} k|_{d_{R_{T1}4}+16} \\ &= h_{PAD_1 1} h_{PAD_1 2} h_{PAD_1 3} h_{PAD_1 4} \end{aligned}$$

In this version of PadGen function, which is known as the simple version, 8 bits out of 16 bits of the resulted PAD_x are decided by R_{Tx}, i.e., the bits that are used to determine $h_{PAD_1 2}$ and $h_{PAD_1 4}$. To provide a better security, HYCLT also introduces a more complex approach to manipulate R_{Tx} and R_{Mx} on the output of PAD_x. Given these definitions and $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ the complex version of PadGen is calculated as follows:

$$\begin{aligned} PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = {} & \mathcal{X}|_{d_{\mathcal{Y}1}+d_{\mathcal{Z}1}} \mathcal{X}|_{d_{\mathcal{Y}1}+d_{\mathcal{Z}2}} \mathcal{X}|_{d_{\mathcal{Y}1}+d_{\mathcal{Z}3}} \mathcal{X}|_{d_{\mathcal{Y}1}+d_{\mathcal{Z}4}} \| \mathcal{X}|_{d_{\mathcal{Y}2}+d_{\mathcal{Z}1}} \mathcal{X}|_{d_{\mathcal{Y}2}+d_{\mathcal{Z}2}} \\ & \mathcal{X}|_{d_{\mathcal{Y}2}+d_{\mathcal{Z}3}} \mathcal{X}|_{d_{\mathcal{Y}2}+d_{\mathcal{Z}4}} \| \mathcal{X}|_{d_{\mathcal{Y}3}+d_{\mathcal{Z}1}} \mathcal{X}|_{d_{\mathcal{Y}3}+d_{\mathcal{Z}2}} \mathcal{X}|_{d_{\mathcal{Y}3}+d_{\mathcal{Z}3}} \mathcal{X}|_{d_{\mathcal{Y}3}+d_{\mathcal{Z}4}} \\ & \| \mathcal{X}|_{d_{\mathcal{Y}4}+d_{\mathcal{Z}1}} \mathcal{X}|_{d_{\mathcal{Y}4}+d_{\mathcal{Z}2}} \mathcal{X}|_{d_{\mathcal{Y}4}+d_{\mathcal{Z}3}} \mathcal{X}|_{d_{\mathcal{Y}4}+d_{\mathcal{Z}4}} \end{aligned}$$

For example assume that

$$\begin{aligned} \mathcal{X} &= 1001\ 1111\ 0011\ 1011\ 0000\ 0011\ 1100\ 0101 \\ \mathcal{Y} &= 0111\ 0100\ 0110\ 1011 \\ \mathcal{Z} &= 1101\ 0110\ 1000\ 1001 \end{aligned}$$

then $PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is calculated as follows:

$$\begin{aligned} PadGen(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) &= \mathcal{X}|_{7+13} \mathcal{X}|_{7+6} \mathcal{X}|_{7+8} \mathcal{X}|_{7+9} \| \mathcal{X}|_{4+13} \mathcal{X}|_{4+6} \mathcal{X}|_{4+8} \mathcal{X}|_{4+9} \| \\ &\quad \mathcal{X}|_{6+13} \mathcal{X}|_{6+6} \mathcal{X}|_{6+8} \mathcal{X}|_{6+9} \| \mathcal{X}|_{11+13} \mathcal{X}|_{11+6} \mathcal{X}|_{11+8} \mathcal{X}|_{11+9} \\ &= \mathcal{X}|_{20} \mathcal{X}|_{13} \mathcal{X}|_{15} \mathcal{X}|_{16} \| \mathcal{X}|_{17} \mathcal{X}|_{10} \mathcal{X}|_{12} \mathcal{X}|_{13} \| \\ &\quad \mathcal{X}|_{19} \mathcal{X}|_{12} \mathcal{X}|_{14} \mathcal{X}|_{15} \| \mathcal{X}|_{24} \mathcal{X}|_{17} \mathcal{X}|_{19} \mathcal{X}|_{20} \\ &= 0010\ 0110\ 0111\ 1000 \end{aligned}$$

Given R_{Tx} and R_{Mx} for $x \in \{1, 2, 3, 4\}$, the new PadGen function is used to generate PAD_x as follows:

$$\begin{aligned} R_{Vx} &= PadGen(APwd, R_{Tx}, R_{Mx}) \\ &= a|_{w|_1} a|_{w|_2} a|_{w|_3} a|_{w|_4} \| a|_{w|_5} a|_{w|_6} a|_{w|_7} a|_{w|_8} \| a|_{w|_9} a|_{w|_{10}} a|_{w|_{11}} a|_{w|_{12}} \\ &\quad \| a|_{w|_{13}} a|_{w|_{14}} a|_{w|_{15}} a|_{w|_{16}} \\ &= d_{R_{Vx}1} d_{R_{Vx}2} d_{R_{Vx}3} d_{R_{Vx}4} \end{aligned}$$

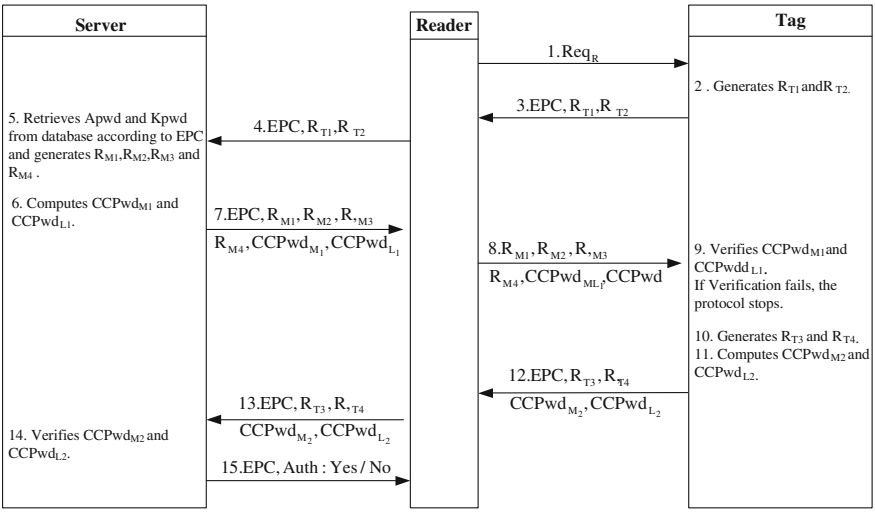

Fig. 1. The mutual authentication protocol proposed by HYCLT et al.

and

$$\begin{aligned} PAD_x &= PadGen(Kpwd, R_{Vx}, R_{Tx}) \\ &= k|_{z|_1} k|_{z|_2} k|_{z|_3} k|_{z|_4} \|k|_{z|_5} k|_{z|_6} k|_{z|_7} k|_{z|_8} \|k|_{z|_9} k|_{z|_{10}} k|_{z|_{11}} k|_{z|_{12}} \\ &\quad \|k|_{z|_{13}} k|_{z|_{14}} k|_{z|_{15}} k|_{z|_{16}} \\ &= h_{PAD_x 1} h_{PAD_x 2} h_{PAD_x 3} h_{PAD_x 4} \end{aligned}$$

where

$$\begin{aligned} w|_{1\sim4} &= d_{R_{Tx}1} + d_{R_{Mx}1}, d_{R_{Tx}1} + d_{R_{Mx}2}, d_{R_{Tx}1} + d_{R_{Mx}3}, d_{R_{Tx}1} + d_{R_{Mx}4} \\ w|_{5\sim8} &= d_{R_{Tx}2} + d_{R_{Mx}1}, d_{R_{Tx}2} + d_{R_{Mx}2}, d_{R_{Tx}2} + d_{R_{Mx}3}, d_{R_{Tx}2} + d_{R_{Mx}4} \\ w|_{9\sim12} &= d_{R_{Tx}3} + d_{R_{Mx}1}, d_{R_{Tx}3} + d_{R_{Mx}2}, d_{R_{Tx}3} + d_{R_{Mx}3}, d_{R_{Tx}3} + d_{R_{Mx}4} \\ w|_{13\sim16} &= d_{R_{Tx}4} + d_{R_{Mx}1}, d_{R_{Tx}4} + d_{R_{Mx}2}, d_{R_{Tx}4} + d_{R_{Mx}3}, d_{R_{Tx}4} + d_{R_{Mx}4} \\ z|_{1\sim4} &= d_{R_{Tx}1} + d_{R_{Vx}1}, d_{R_{Tx}1} + d_{R_{Vx}2}, d_{R_{Tx}1} + d_{R_{Vx}3}, d_{R_{Tx}1} + d_{R_{Vx}4} \\ z|_{5\sim8} &= d_{R_{Tx}2} + d_{R_{Vx}1}, d_{R_{Tx}2} + d_{R_{Vx}2}, d_{R_{Tx}2} + d_{R_{Vx}3}, d_{R_{Tx}2} + d_{R_{Vx}4} \\ z|_{9\sim12} &= d_{R_{Tx}3} + d_{R_{Vx}1}, d_{R_{Tx}3} + d_{R_{Vx}2}, d_{R_{Tx}3} + d_{R_{Vx}3}, d_{R_{Tx}3} + d_{R_{Vx}4} \\ z|_{13\sim16} &= d_{R_{Tx}4} + d_{R_{Vx}1}, d_{R_{Tx}4} + d_{R_{Vx}2}, d_{R_{Tx}4} + d_{R_{Vx}3}, d_{R_{Tx}4} + d_{R_{Vx}4} \end{aligned}$$

A more detailed description of HYCLT protocol is provided in Fig. 1 which is described as below:

1. The reader starts the protocol by sending Req_R to the tag.
2. On reception, the tag generates two random numbers R_{T1} and R_{T2} and sends its EPC with R_{T1} and R_{T2} to the reader.

3. Once the reader receipts this message, it forwards the message to the server.
4. Upon receipt the message, the server:
   - retrieves $Apwd$ and $Kpwd$ from database according to EPC;
   - generates four fresh random numbers R_{M1}, R_{M2}, R_{M3} and R_{M4};
   - computes $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $CCPwd_{L_1} = Apwd_L \oplus PAD_2$;
   - sends $EPC, R_{M1}, R_{M2}, R_{M3}, R_{M4}, CCPwd_{M_1}$ and $CCPwd_{L_1}$ to the reader.
5. Upon receipt of the message, the reader sends R_{M1}, R_{M2}, R_{M3}, R_{M4}, $CCPwd_{M_1}$ and $CCPwd_{L_1}$ to the tag.
6. Upon receipt of the message, the tag verifies the correctness of $CCPwd_{M_1}$ and $CCPwd_{L_1}$ and does as follows:
   - generates R_{T3} and R_{T4};
   - computes $CCPwd_{M_2} = Apwd_M \oplus PAD_3$ and $CCPwd_{L_2} = Apwd_L \oplus PAD_4$;
   - and sends EPC, R_{T3}, R_{T4}, $CCPwd_{M_2}$ and $CCPwd_{L_2}$ to the reader.
7. The reader forwards the message to the server.
8. The server verifies $CCPwd_{M_2}$ and $CCPwd_{L_2}$. If they are valid it sends EPC and $Auth : Yes$ to the reader; Otherwise, it sends EPC and $Auth : No$ to the reader.

2.1 Secret Disclosure Attack on HYCLT Protocol

Considering HYCLT based on its complex PadGen function, in this section we present an attack which retrieves the secret *Access* password of any given tag in HYCLT. The presented attack is based on the following observation:

Observation 1: Assume that in Step 2 of the protocol, where the reader has started the protocol by sending Req_R to the tag and the tag generates two random numbers R_{T1} and R_{T2} and sends its EPC with R_{T1} and R_{T2} to the reader, the adversary intercepts R_{T1} and R_{T2} sent by the tag and replaces them by R'_{T1} and R'_{T2} such that, e.g., $R'_{T1} = d_{R'_{T1}1} \| d_{R'_{T1}1} \| d_{R'_{T1}1} \| d_{R'_{T1}1}$ and $R'_{T2} = d_{R'_{T2}1} \| d_{R'_{T2}1} \| d_{R'_{T2}1} \| d_{R'_{T2}1}$ where $d_{R'_{T1}1}$ or $d_{R'_{T2}1}$ could be any value $\in \{0, \ldots, 15\}$. An example is $R'_{T1} = R'_{T2} = 0$. Then we have $w|_{1\sim4} = w|_{5\sim8} = w|_{9\sim12} = w|_{13\sim16}$ and equivalently we can state that $d_{R_{Vx}1} = d_{R_{Vx}2} = d_{R_{Vx}3} = d_{R_{Vx}4}$. Consider $x = 1$, we have $d_{R_{V1}1} = d_{R_{V1}2} = d_{R_{V1}3} = d_{R_{V1}4}$ and $d_{R'_{T1}1} = d_{R'_{T1}2} = d_{R'_{T1}3} = d_{R'_{T1}4}$ On the other hand,

$$\begin{aligned} PAD_1 &= PadGen(Kpwd, R_{V1}, R'_{T1}) \\ &= k|_{z1}k|_{z2}k|_{z3}k|_{z4} \| k|_{z5}k|_{z6}k|_{z7}k|_{z8} \| k|_{z9}k|_{z10}k|_{z11}k|_{z12} \| k|_{z13}k|_{z14}k|_{z15}k|_{z16} \\ &= h_{PAD_1 1} h_{PAD_1 2} h_{PAD_1 3} h_{PAD_1 4} \end{aligned}$$

where

$$\begin{aligned} z|_{1\sim4} &= d_{R'_{T1}1} + d_{R_{V1}1}, d_{R'_{T1}1} + d_{R_{V1}2}, d_{R'_{T1}1} + d_{R_{V1}3}, d_{R'_{T1}1} + d_{R_{V1}4} \\ z|_{5\sim8} &= d_{R'_{T1}2} + d_{R_{V1}1}, d_{R'_{T1}2} + d_{R_{V1}2}, d_{R'_{T1}2} + d_{R_{V1}3}, d_{R'_{T1}2} + d_{R_{V1}4} \\ z|_{9\sim12} &= d_{R'_{T1}3} + d_{R_{V1}1}, d_{R'_{T1}3} + d_{R_{V1}2}, d_{R'_{T1}3} + d_{R_{V1}3}, d_{R'_{T1}3} + d_{R_{V1}4} \\ z|_{13\sim16} &= d_{R'_{T1}4} + d_{R_{V1}1}, d_{R'_{T1}4} + d_{R_{V1}2}, d_{R'_{T1}4} + d_{R_{V1}3}, d_{R'_{T1}4} + d_{R_{V1}4}. \end{aligned}$$

Since $d_{R'_{T_1}i} + d_{R_{V1}j}$ for any i, and $j \in \{1,2,3,4\}$ is a fixed value here, we have $z|_m = z|_n$ for any m and $n \in \{1,\ldots,16\}$. Therefore we have $z|_1 = z|_2 = \ldots = z|_{16} = z$. Hence $PAD_1 = PadGen(Kpwd, R_{V1}, R'_{T1}) = k|_z \| k|_z \| \ldots \| k|_z$ where $z \in \{0,\ldots,F\}$ and $PAD_1 = PadGen(Kpwd,\ R_{V1},\ R'_{T1}) \in \{0000, FFFF\}$.

Similarly for $x = 2$ we have $d_{R_{V2}1} = d_{R_{V2}2} = d_{R_{V2}3} = d_{R_{V2}4}$ and $d_{R'_{T_2}1} = d_{R'_{T_2}2} = d_{R'_{T_2}3} = d_{R'_{T_2}4}$. On the other hand,

$$\begin{aligned} PAD_2 &= PadGen(Kpwd, R_{V2}, R'_{T2}) \\ &= k|_{z'1} k|_{z'2} k|_{z'3} k|_{z'4} \| k|_{z'5} k|_{z'6} k|_{z'7} k|_{z'8} \| k|_{z'9} k|_{z'10} k|_{z'11} k|_{z'12} \\ &\quad \| k|_{z'13} k|_{z'14} k|_{z'15} k|_{z'16} \\ &= h_{PAD_21} h_{PAD_22} h_{PAD_23} h_{PAD_24} \end{aligned}$$

where

$$\begin{aligned} z'_{1\sim4} &= d_{R'_{T_2}1} + d_{R_{V2}1}, d_{R'_{T_2}1} + d_{R_{V2}2}, d_{R'_{T_2}1} + d_{R_{V2}3}, d_{R'_{T_2}1} + d_{R_{V2}4} \\ z'_{5\sim8} &= d_{R'_{T_2}2} + d_{R_{V2}1}, d_{R'_{T_2}2} + d_{R_{V2}2}, d_{R'_{T_2}2} + d_{R_{V2}3}, d_{R'_{T_2}2} + d_{R_{V2}4} \\ z'_{9\sim12} &= d_{R'_{T_2}3} + d_{R_{V2}1}, d_{R'_{T_2}3} + d_{R_{V2}2}, d_{R'_{T_2}3} + d_{R_{V2}3}, d_{R'_{T_2}3} + d_{R_{V2}4} \\ z'_{13\sim16} &= d_{R'_{T_2}4} + d_{R_{V2}1}, d_{R'_{T_2}4} + d_{R_{V2}2}, d_{R'_{T_2}4} + d_{R_{V2}3}, d_{R'_{T_2}4} + d_{R_{V2}4}. \end{aligned}$$

Since $d_{R'_{T_2}i} + d_{R_{V2}j}$ for any i and $j \in \{1,2,3,4\}$ is a fixed value here, we have $z'_m = z'_n$ for any m and $n \in \{1,\ldots,16\}$. Therefore we have $z'_1 = z'_2 = \ldots = z'_{16} = z'$. Hence $PAD_2 = PadGen(Kpwd, R_{V2}, R'_{T2}) = k|'_z \| k|'_z \| \ldots \| k|_{z'}$ where $z' \in \{0,\ldots,F\}$ and $PAD_2 = PadGen(Kpwd,\ R_{V2},\ R'_{T2}) \in \{0000, FFFF\}$.

Now given that $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $CCPwd_{L_1} = Apwd_L \oplus PAD_2$, and there are two choices for any of PAD_1 and PAD_2 (in total 4 choices) the adversary can determine the correct $Apwd_L \| Apwd_M$ with the probability of 2^{-2}, where $Apwd = a|_0 a|_1 a|_2 a|_3 \ldots a|_{31}$, $Apwd_L = a|_0 a|_1 \ldots a|_{15}$ and $Apwd_M = a|_{16} a|_{17} \ldots a|_{31}$.

Following **observation 1**, we have $z|_1 = z|_2 = \ldots = z|_{16} = z$. Hence $PAD_x = PadGen(Kpwd, R_{Vx}, R'_{Tx}) = k|_z \| k|_z \| \ldots \| k|_z$ where $z \in \{0,\ldots,F\}$ and $PAD_x = PadGen\ (Kpwd, R_{Vx}, R'_{Tx}) \in \{0000, FFFF\}$. Now given that $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $CCPwd_{L_1} = Apwd_L \oplus PAD_2$, the adversary can determine $Apwd_L \| Apwd_M$ with the probability of 2^{-2}, where $Apwd = a|_0 a|_1 a|_2 a|_3 \ldots a|_{31}$, $Apwd_M = a|_0 a|_1 \ldots a|_{15}$ and $Apwd_L = a|_{16} a|_{17} \ldots a|_{31}$.

3 HLL Protocol

Huang, Lin and Li in [6] have presented another EPC- C1 G2 specification complaint mutual authentication protocol with a different PadGen function and successfully verified their protocol functionality in hardware. We refer to this protocol by HLL.

In the PadGen function of the simple variant of HYCLT protocol and Konidala et al. protocol the location of a fraction of the bits of secret passwords that are included in PAD_x are decided by a public parameter which is under the adversary's control. For example, in the simple variant of HYCLT protocol, PAD_1 is calculated as $PAD_1 = PadGen(Kpwd, R_{V1}, R_{T1})$, where the

location of the extraction of 8 bits out of 16 bits are determined by R_{T_1} which is under the adversary's control. The PadGen function proposed in HLL protocol is computed based on a set of values, i.e., (R_{Vx}, R_{Wx}), which is calculated inside the server or tags and they are not transmitted over the channel between the reader and the tag. There are two variants of PadGen function in HLL protocol based on XOR or MOD operation respectively. In this paper we consider the variant based on XOR and the presented attack does not work for the protocol based on MOD. To calculate PAD_1 and PAD_2 values, given two random number R_{Tx} and R_{Mx} that are generated by the tag and the server respectively, at the first an intermediate parameter denoted by $R_{Tx\oplus Mx}$ is calculated as $R_{Tx\oplus Mx} = R_{Tx} \oplus R_{Mx} = d_{R_{Tx\oplus Mx}1} d_{R_{Tx\oplus Mx}2} d_{R_{Tx\oplus Mx}3} d_{R_{Tx\oplus Mx}4}$. This parameter is used as an input for the PadGen function to calculate another temporary value denoted by R_{Wx} as follows:

$$\begin{aligned} R_{Wx} &= PadGen(APwd, R_{Tx}, R_{Tx\oplus Mx}) \\ &= a|_{d_{R_{Tx}1}} a|_{d_{R_{Tx}2}} a|_{d_{R_{Tx}3}} a|_{d_{R_{Tx}4}} \| a|_{d_{R_{Tx}1}+16} a|_{d_{R_{Tx}2}+16} a|_{d_{R_{Tx}3}+16} a|_{d_{R_{Tx}4}+16} \\ &\quad \| a|_{d_{R_{Tx\oplus Mx}1}} a|_{d_{R_{Tx\oplus Mx}2}} a|_{d_{R_{Tx\oplus Mx}3}} a|_{d_{R_{Tx\oplus Mx}4}} \\ &\quad \| a|_{d_{R_{Tx\oplus Mx}1}+16} a|_{d_{R_{Tx\oplus Mx}2}+16} a|_{d_{R_{Tx\oplus Mx}3}+16} a|_{d_{R_{Tx\oplus Mx}4}+16}. \end{aligned}$$

In addition, R_{Tx} and R_{Mx} are used as the input of PadGen function to calculate a temporary value R_{Vx} as follows:

$$\begin{aligned} R_{Vx} &= PadGen(APwd, R_{Tx}, R_{Mx}) \\ &= a|_{d_{R_{Tx}1}} a|_{d_{R_{Tx}2}} a|_{d_{R_{Tx}3}} a|_{d_{R_{Tx}4}} \| a|_{d_{R_{Tx}1}+16} a|_{d_{R_{Tx}2}+16} a|_{d_{R_{Tx}3}+16} a|_{d_{R_{Tx}4}+16} \| \\ &\quad a|_{d_{R_{Mx}1}} a|_{d_{R_{Mx}2}} a|_{d_{R_{Mx}3}} a|_{d_{R_{Mx}4}} \| a|_{d_{R_{Mx}1}+16} a|_{d_{R_{Mx}2}+16} a|_{d_{R_{Mx}3}+16} a|_{d_{R_{Mx}4}+16} \end{aligned}$$

Given R_{W1} and R_{V1}, PAD_1 function is calculated as follows:

$$\begin{aligned} PAD_1 &= PadGen(Kpwd, R_{V1}, R_{W1}) \\ &= k|_{d_{R_{V1}1}} k|_{d_{R_{V1}2}} k|_{d_{R_{V1}3}} k|_{d_{R_{V1}4}} \| k|_{d_{R_{V1}1}+16} k|_{d_{R_{V1}2}+16} k|_{d_{R_{V1}3}+16} k|_{d_{R_{V1}4}+16} \| \\ &\quad k|_{d_{R_{W1}1}} k|_{d_{R_{W1}2}} k|_{d_{R_{W1}3}} k|_{d_{R_{W1}4}} \| k|_{d_{R_{W1}1}+16} k|_{d_{R_{W1}2}} \\ &\quad + 16k|_{d_{R_{W1}3}+16} k|_{d_{R_{W1}4}+16} \end{aligned}$$

To calculate PAD_2, the protocol at the first calculates a new parameter $R_{V1\oplus W1}$ as $R_{S1} = R_{V1\oplus W1} = R_{V1} \oplus R_{W1}$. Given R_{S1} and R_{V1}, the value of PAD_2 is calculated as follows:

$$\begin{aligned} PAD_2 &= PadGen(Kpwd, R_{V1}, R_{S1}) \\ &= k|_{d_{R_{V1}1}} k|_{d_{R_{V1}2}} k|_{d_{R_{V1}3}} k|_{d_{R_{V1}4}} \| k|_{d_{R_{V1}1}+16} k|_{d_{R_{V1}2}+16} k|_{d_{R_{V1}3}+16} k|_{d_{R_{V1}4}+16} \| \\ &\quad k|_{d_{R_{S1}1}} k|_{d_{R_{S1}2}} k|_{d_{R_{S1}3}} k|_{d_{R_{S1}4}} \| k|_{d_{R_{S1}1}+16} k|_{d_{R_{S1}2}+16} k|_{d_{R_{S1}3}+16} k|_{d_{R_{S1}4}+16} \end{aligned}$$

A description of HLL protocol is provided in Fig. 2 which is described as follows:

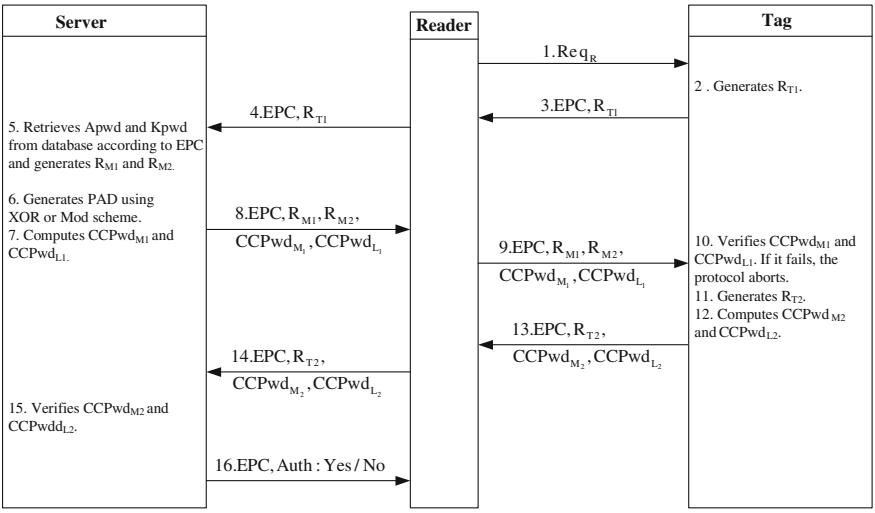

Fig. 2. The Huang-Lin-Li mutual authentication protocol using XOR or MOD scheme.

1. The reader starts the protocol by sending Req_R to the tag.
2. On reception, the tag generates a random number R_{T1} and sends its EPC with R_{T1} to the reader.
3. Once the reader receipt the message, forwards it to the server.
4. Upon receipt the message, the server :
   - retrieves $Apwd$ and $Kpwd$ from database according to EPC;
   - generates two random numbers R_{M1} and R_{M2};
   - generates PAD using XOR or MOD scheme, where we concentrate on XOR operation.
   - computes $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $CCPwd_{L_1} = Apwd_L \oplus PAD_2$;
   - and sends EPC, R_{M1}, R_{M2}, $CCPwd_{M_1}$ and $CCPwd_{L_1}$ to the reader.
5. On receipt the message, the reader forwards the message to the tag.
6. Upon receipt the message, the tag verifies $CCPwd_{M_1}$ and $CCPwd_{L_1}$. If the equality does not exist, the protocol will stop. Otherwise it:
   - generates another random number R_{T2};
   - computes $CCPwd_{M_2} = Apwd_M \oplus PAD_3$ and $CCPwd_{L_2} = Apwd_L \oplus PAD_4$;
   - and sends $EPC, R_{T2}, CCPwd_{M_2}$ and $CCPwd_{L_2}$ to the reader.
7. The reader forwards the message to the server.
8. The server verifies $CCPwd_{M_2}$ and $CCPwd_{L_2}$. In the case of equality, sends EPC and $Auth : Yes$ to the reader. Otherwise it sends EPC and $Auth : No$ to the reader.

3.1 Security Analysis of the HLL Protocol

Passive Adversary

Observation 1: It can be seen that $d_{R_{V}1}1 = d_{R_{W}1}1 = a|_{d_{R_{T}1}1}a|_{d_{R_{T}1}2}$ $a|_{d_{R_{T}1}3}a|_{d_{R_{T}1}4}$ and $d_{R_{V}1}2 = d_{R_{W}1}2 = a|_{d_{R_{T}1}1+16}a|_{d_{R_{T}1}2+16}a|_{d_{R_{T}1}3+16}a|_{d_{R_{T}1}4+16}$.

Observation 2: Following **Observation 1**, $k|_{d_{R_{V}1}1} = k|_{d_{R_{W}1}1}$, $k|_{d_{R_{V}1}2} = k|_{d_{R_{W}1}2}$, $k|_{d_{R_{V}1}1+16} = k|_{d_{R_{W}1}1+16}$ and $k|_{d_{R_{V}1}2+16} = k|_{d_{R_{W}1}2+16}$.

Following this observation PAD_1 can be rewritten as follows:

$$PAD_1 = k|_{d_{R_{V}1}1}k|_{d_{R_{V}1}2}k|_{d_{R_{V}1}3}k|_{d_{R_{V}1}4}\|k|_{d_{R_{V}1}1+16}k|_{d_{R_{V}1}2+16}k|_{d_{R_{V}1}3+16}k|_{d_{R_{V}1}4+16}\|$$
$$k|_{d_{R_{V}1}1}k|_{d_{R_{V}1}2}k|_{d_{R_{W}1}3}k|_{d_{R_{W}1}4}\|k|_{d_{R_{V}1}1+16}k|_{d_{R_{V}1}2+16}k|_{d_{R_{W}1}3+16}k|_{d_{R_{W}1}4+16}.$$

Given that $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $Apwd_M = a|_{16}a|_{17}\ldots a|_{31}$, we can extract the following equations:

$$(CCPwd_{M_1})|_0 \oplus (CCPwd_{M_1})|_8 = a|_{16} \oplus a|_{24}$$
$$(CCPwd_{M_1})|_1 \oplus (CCPwd_{M_1})|_9 = a|_{17} \oplus a|_{25}$$
$$(CCPwd_{M_1})|_4 \oplus (CCPwd_{M_1})|_{12} = a|_{20} \oplus a|_{28}$$
$$(CCPwd_{M_1})|_5 \oplus (CCPwd_{M_1})|_{13} = a|_{21} \oplus a|_{29};$$

which can be used to recognize a target tag with the success probability of $1 - 2^{-4}$.

Observation 3: Following **observation 1**, one can state that $d_{R_{S}1}1 = d_{R_{S}1}2 = 0$ and $R_{V1\oplus W1} = 00d_{R_{S}1}3d_{R_{S}1}4$.

On the other hand:

$$PAD_2 = k|_{d_{R_{V}1}1}k|_{d_{R_{V}1}2}k|_{d_{R_{V}1}3}k|_{d_{R_{V}1}4}\|k|_{d_{R_{V}1}1+16}k|_{d_{R_{V}1}2+16}k|_{d_{R_{V}1}3+16}k|_{d_{R_{V}1}4+16}\|$$
$$k|_{d_{R_{S}1}1}k|_{d_{R_{S}1}2}k|_{d_{R_{S}1}3}k|_{d_{R_{S}1}4}\|k|_{d_{R_{S}1}1+16}k|_{d_{R_{S}1}2+16}k|_{d_{R_{S}1}3+16}k|_{d_{R_{S}1}4+16}$$

Hence, we can rewrite PAD_2 as follows:

$$PAD_2 = k|_{d_{R_{V}1}1}k|_{d_{R_{V}1}2}k|_{d_{R_{V}1}3}k|_{d_{R_{V}1}4}\|k|_{d_{R_{V}1}1+16}k|_{d_{R_{V}1}2+16}k|_{d_{R_{V}1}3+16}k|_{d_{R_{V}1}4+16}\|$$
$$k|_0k|_0k|_{d_{R_{S}1}3}k|_{d_{R_{S}1}4}\|k|_{16}k|_{16}k|_{d_{R_{S}1}3+16}k|_{d_{R_{S}1}4+16}.$$

So, $CCPwd_{L_1} = xxxx\|xxxx\|(k|_0 \oplus a|_8)(k|_0 \oplus a|_9)xx\|(k|_{16} \oplus a|_{12})(k|_{16} \oplus a|_{13})xx$, which can be used to recognize a target tag with the success probability of $1-2^{-4}$. This information also leaks 4 bits of secret passwords.

Observation 4: Comparing the details of PAD_1 and PAD_2 we can see that $h_{PAD_1}1 = h_{PAD_2}1$ and $h_{PAD_1}2 = h_{PAD_2}2$. Now given that $CCPwd_{M_1} = Apwd_M \oplus PAD_1$ and $CCPwd_{L_1} = Apwd_L \oplus PAD_2$, the adversary can use the 8-LSB of $CCPwd_{M_1} \oplus CCPwd_{L_1}$ as a measure to trace the target tag, which is independent of the nonces and only dependent on the $Apwd_L \oplus Apwd_M$ and is static.

More precisely (x denotes an unknown binary value):

$$\begin{aligned}
PAD_1 &= h_{PAD_1 1} h_{PAD_1 2} h_{PAD_1 3} h_{PAD_1 4} \\
PAD_2 &= h_{PAD_2 1} h_{PAD_2 2} h_{PAD_2 3} h_{PAD_2 4} \\
CCPwd_{M_1} &= Apwd_M \oplus PAD_1 \\
CCPwd_{L_1} &= Apwd_L \oplus PAD_2 \\
PAD_1 \oplus PAD_2 &= 0000 \| 0000 \| xxxx \| xxxx; (Observation\ 4) \\
Apwd_L &= a|_0 a|_1 \ldots a|_{15} \\
Apwd_M &= a|_{16} a|_{17} \ldots a|_{31} \\
CCPwd_{L_1} \oplus CCPwd_{M_1} &= (a|_0 \oplus a|_{16})(a|_1 \oplus a|_{17})(a|_2 \oplus a|_{18})(a|_3 \oplus a|_{19}) \| \\
&\quad (a|_4 \oplus a|_{20})(a|_5 \oplus a|_{21})(a|_6 \oplus a|_{22})(a|_7 \oplus a|_{23}) \\
&\quad \| xxxx \| xxxx.
\end{aligned}$$

Active Adversary. Assume that an active adversary intercepts the message from the tag to the reader in step 3 and replaces R_{T1} by $R^i_{T1} = d_{R^i_{T_1}1} \| d_{R^i_{T_1}2} \| d_{R^i_{T_1}3} \| d_{R^i_{T_1}4} = i \| i \| i \| i$, for $0 \geq i \geq 15$. Then, one can state that $d_{R^i_{V_1}1} = d_{R^i_{W_1}1} = a|_i a|_i a|_i a|_i \in \{0000, 1111\}$ (base 2). In addition, we assume that $k|_0 \oplus k|_{15} = k|_{16} \oplus k|_{31} = 1$. Now, given these assumptions and given $CCPwd^i_{M_1}$ and $CCPwd^j_{M_1}$, we can find out whether $a|_i = a|_j$ as follows:

$$\begin{aligned}
&CCPwd^i_{M_1} \oplus CCPwd^j_{M_1} \\
&= Apwd_M \oplus PAD^i_1 \oplus Apwd_M \oplus PAD^j_1 \\
&= PAD^i_1 \oplus PAD^j_1 \\
&= (k|_{d_{R^i_{V1}1}} \oplus k|_{d_{R^j_{V1}1}})(k|_{d_{R^i_{V1}2}} \oplus k|_{d_{R^j_{V1}2}})(k|_{d_{R^i_{V1}3}} \oplus k|_{d_{R^j_{V1}3}})(k|_{d_{R^i_{V1}4}} \oplus k|_{d_{R^j_{V1}4}}) \| \\
&\quad (k|_{d_{R^i_{V1}1}+16} \oplus k|_{d_{R^j_{V1}1}+16})(k|_{d_{R^i_{V1}2}+16} \oplus k|_{d_{R^j_{V1}2}+16})(k|_{d_{R^i_{V1}3}+16} \oplus k|_{d_{R^j_{V1}3}+16}) \\
&\quad (k|_{d_{R^i_{V1}4}+16} \oplus k|_{d_{R^j_{V1}4}+16}) \| (k|_{d_{R^i_{W1}1}} \oplus k|_{d_{R^j_{W1}1}})(k|_{d_{R^i_{W1}2}} \oplus k|_{d_{R^j_{W1}2}})(k|_{d_{R^i_{W1}3}} \\
&\quad \oplus k|_{d_{R^j_{W1}3}})(k|_{d_{R^i_{W1}4}} \oplus k|_{d_{R^j_{W1}4}}) \| (k|_{d_{R^i_{W1}1}+16} \oplus k|_{d_{R^j_{W1}1}+16})(k|_{d_{R^i_{W1}2}+16} \\
&\quad \oplus k|_{d_{R^j_{W1}2}+16})(k|_{d_{R^i_{W1}3}+16} \oplus k|_{d_{R^j_{W1}3}+16})(k|_{d_{R^i_{W1}4}+16} \oplus k|_{d_{R^j_{W1}4}+16}).
\end{aligned}$$

Since, $k|_0 \oplus k|_{15} = 1$ and $d_{R^i_{V1}1} \in \{0, 15\}$ and $d_{R^j_{V1}1} \in \{0, 15\}$ then $(k|_{d_{R^i_{V1}1}} \oplus k|_{d_{R^j_{V1}1}}) = 0$ implies that $a|_i = a|_j$ and vice versa. Hence the adversary can fix $j = 0$ and varies i from 1 to 15 and verifies whether $a|_i = a|_0$. In this way the adversary can determine all bits of $Apwd_L$ with the success probability of $\frac{1}{2}$. Following the same approach for $(k|_{d_{R^i_{V1}2}} \oplus k|_{d_{R^j_{V1}2}})$ all bits of $Apwd_M$ can be determined with the success probability of $\frac{1}{2}$. Hence an active adversary can determine all 32 bits of $Apwd$ with success probability of 2^{-4}. On the other hand, given $Apwd$, R_{Tx} and R_{Mx} the adversary can determine PAD_1, PAD_2, R_{V1}, R_{W1} and $R_{V1 \oplus W1}$. Given this information, the adversary can also retrieve $Kpwd$.

4 Conclusion

In this paper we considered the security of two RFID mutual authentication protocols conforming to the EPC-C1 G2 standard. In these two protocols, authors aimed to solve ISO 18000-6C protocol weaknesses by using a special pad generation function named PadGen to mask tag's *Access* password $Apwd = Apwd_M \| Apwd_L$ before the data is transmitted. We showed that an attacker can obtain the *Access* and *Kill* passwords with high probability. We found that in Huang et al. scheme the adversary can determine the *Access* password with the probability of 2^{-2}, and in Huang-Lin-Li scheme the passive adversary can trace a target tag with the success probability of $1 - 2^{-4}$ and the active adversary can determine all 32 bits of *Access* password with success probability of 2^{-4}. Given this information, the adversary can also retrieve $Kpwd$. By knowing *Access* and *Kill* passwords the attacker can access the tag's memory and can make the target inoperative respectively.

Acknowledgments. We would like to thank anonymous reviewers for useful comments.

References

1. Bailey, D.V., Juels, A.: Shoehorning security into the EPC tag standard. In: De Prisco, R., Yung, M. (eds.) SCN 2006, LNCS, vol. 4116, pp. 303–320. Springer, Heidelberg (2006)
2. Chen, C.-L., Chien, C.-F.: Based on mobile RFID for membership stores system conforming EPC C1 G2 standards. IJAHUC **10**(4), 207–218 (2012)
3. Chen, C.-L., Deng, Y.-Y.: Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection. Eng. Appl. AI **22**(8), 1284–1291 (2009)
4. Chien, H.-Y., Chen, C.-H.: Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. Comput. Stand. Interfaces **29**(2), 254–259 (2007)
5. EPCGlobal: Class-1 generation 2 UHF air interface protocol standard version 1.2.0, Gen2 Standard. http://www.epcglobalinc.org/standards/ (2008)
6. Huang, Y.-J., Lin, W.-C., Li, H.-L.: Efficient implementation of RFID mutual authentication protocol. IEEE Trans. Industr. Electron. **59**(12), 4784–4791 (2012)
7. Huang, Y.-J., Yuan, C.-C., Chen, M.-K., Lin, W.-C., Teng, H.-C.: Hardware implementation of RFID mutual authentication protocol. IEEE Trans. Industr. Electron. **57**(5), 1573–1582 (2010)
8. Information technology - Radio frequency identification for item management. Part 6: Parameters for air interface communications at 860 MHz to 960 MHz. http://www.iso.org/iso/catalogue_detail?csnumber=34117 (2005)
9. Konidala, D., Kim, Z., Kim, K.: A simple and cost effective RFID tag-reader mutual authentication scheme. In: Proceedings of International Conference on RFID Security, pp. 141–152, July 2007
10. Ma, C., Li, Y., Deng, R.H., Li, T.: RFID privacy: relation between two notions, minimal condition, and efficient construction. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM Conference on Computer and Communications, Security, pp. 54–65. ACM Press, New York (2009)

11. Park, J., Na, J., Kim, M.: A practical approach for enhancing security of EPCglobal RFID Gen2 tag. In: FGCN (1), pp. 436–441. IEEE (2007)
12. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J., Ribagorda, A.: Practical attacks on a mutual authentication scheme under the EPC Class-1 Generation-2 standard. Comput. Commun. **32**(7–10), 1185–1193 (2009)
13. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: RFID specification revisited. In: The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, pp. 311–346. Taylor & Francis Group, London (2008)
14. Want, R.: An introduction to RFID technology. IEEE Pervasive Comput. **5**(1), 25–33 (2006)