Computer Communications 32 (2009) 1185–1193

Contents lists available at ScienceDirect

Computer Communications

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

# Practical attacks on a mutual authentication scheme under the EPC Class-1 Generation-2 standard

Pedro Peris-Lopez [a,*], Tieyan Li [b], Julio C. Hernandez-Castro [c], Juan M.E. Tapiador [d]

[a] Delft University of Technology (TU-Delft), Faculty of Electrical Engineering, Mathematics, and Computer Science (EEMCS), Information and Communication Theory group (ICT), P.O. Box 5031 2600 GA, Delft, The Netherlands
[b] Institute for Infocomm Research, A*STAR Singapore, Singapore
[c] School of Computing Science, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, United Kingdom
[d] Department of Computer Science, University of York, Heslington, York YO10 5DD, United Kingdom

ARTICLE INFO

*Article history:*
Received 8 October 2008
Received in revised form 21 January 2009
Accepted 20 March 2009
Available online 5 April 2009

*Keywords:*
RFID
Security
Attacks
EPC-C1G2 standard

ABSTRACT

The EPC Class-1 Generation-2 RFID standard provides little security, as has been shown in previous works such as [S. Karthikeyan, M. Nesterenko, RFID security without extensive cryptography, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, 2005, pp. 63–67; D.N. Duc, J. Park, H. Lee, K. Kim, Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning, in: The 2006 Symposium on Cryptography and Information Security, 2006; H.Y. Chien, C.H. Chen, Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards, Computer Standards & Interfaces 29 (2007) 254–259; P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, A. Ribagorda, Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard, in: Proceedings of Int'l Conference on RFID Security (RFIDSec)'07, Jul 2007; T.L. Lim, T. Li, Addressing the weakness in a lightweight RFID tag-reader mutual authentication scheme, in Proceedings of the IEEE Int'l Global Telecommunications Conference (GLOBECOM) 2007, Nov 2007, pp. 59–63]. In particular, the security of an RFID tag's access and kill passwords is almost non-existent. Konidala and Kim recently proposed a new mutual authentication scheme [D.M. Konidala, Z. Kim, K. Kim, A simple and cost-effective RFID tag-reader mutual authentication scheme, in: Proceedings of Int'l Conference on RFID Security (RFIDSec)'07, Jul 2007, pp. 141–152] – an improved version of their first attempt [D.M. Konidala, K. Kim, RFID tag-reader mutual authentication scheme utilizing tag's access password, Auto-ID Labs White Paper WP-HARDWARE-033, Jan 2007] – in which a tag's access and kill passwords are used for authentication. In this paper, we show that the new scheme continues to present serious security flaws. The 16 least significant bits of the access password can be obtained with probability $2^{-2}$, and the 16 most significant bits with a probability greater than $2^{-5}$. Finally, we show how an attacker can recover the entire kill password with probability $2^{-2}$.



## 1. Introduction

EPCglobal is an organization that develops standards for the electronic product code (EPC), as well as an RFID framework (the EPCglobal Architecture Framework [9]) that supports the use of EPC. Among the few standards that have been developed by EPCglobal is the EPC Class-1 Gen-2 standard (EPC-C1G2 for short) [10] that specifies the RFID communications protocol for ultra-high frequency (UHF) communication between 860 and 960 MHz. The standard specifies that a compliant RFID tag should contain two 32-bit secrets – the access password and the kill password. The access password is used to authenticate readers that wish to access information stored on the tag and controls access to the information. The kill password is used to disable the tag permanently. Once a tag has been 'killed', it is rendered in silence thereafter.

EPC-C1G2 standard additionally specifies a simple scheme that allows a tag to authenticate a reader. This attempts to protect the access password by using a simple form of masking prior to transmission. However, a passive eavesdropper monitoring the messages exchanged between reader and tag can acquire this sensitive information by simply computing an XOR operation [11]. Konidala and Kim exposed this weakness and proposed tag-reader mutual authentication scheme (TRMA) to protect the access password [8]. Unfortunately, their scheme was soon found to be vulnerable to a number of attacks that could reveal the access password [5]. Konidala and Kim therefore proposed a new improved scheme (TRMA$^+$) in [7], which they claimed to be secure. However,

* Corresponding author. Tel.: +34 616 654 680.
*E-mail addresses:* P.PerisLopez@tudelft.nl, pperis@inf.uc3m.es (P. Peris-Lopez), litieyan@i2r.a-star.edu.sg (T. Li), Julio.Hernandez-Castro@port.ac.uk (J.C. Hernandez-Castro), jet@cs.york.ac.uk (J.M.E. Tapiador).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.03.010

under close scrutiny, we have found that there are still weaknesses in the scheme. We show how practical attacks can effectively disclose both the access and the kill passwords. This paper highlights the need for designers of security schemes to conduct stringent cryptanalysis in order to be sure that their schemes provide the necessary resilience to attacks.

Other approaches to RFID authentication that conform to the EPC-C1G2 standard have been proposed previously. An efficient tag identification and reader authentication protocol based on the use of XORs and matrix operations was proposed in [1]. Duc et al. proposed a tag-to-backend database authentication protocol in [2], which uses a 16-bit pseudo-random number generator (PRNG), cyclic redundancy checksum (CRC) and XOR operations. Chien and Chen examined both protocols and pointed out particular security flaws [3]. Correspondingly, they proposed a new scheme similar to that of Duc et al., but this later came under attack by Peris-Lopez et al. [4].

## 2. The original TRMA scheme and its extension

For completeness and coherence we will first provide with a brief description of the original TRMA scheme and its extended version TRMA$^+$.

### 2.1. Original TRMA scheme

In [8], Konidala and Kim proposed an authentication scheme in an attempt to correct the security shortcomings discovered in the EPC-C1G2 standard. They claimed that the proposed scheme protects the tag's access password against disclosure to attackers. A brief description of the TRMA scheme is provided below. For further details, the reader is referred to the original work [8].

**Tag → Reader:** $EPC, RN_1^{Tag}, RN_2^{Tag}$

First, the tag is singulated and backscatters its *EPC* number. Then, the reader sends two *ReqRN* commands to the tag, which responds by backscattering two generated 16-bit random numbers: $RN_1^{Tag}$ and $RN_2^{Tag}$.

**Reader → Tag:** $RN_1^{Rdr}, RN_2^{Rdr}, CCPwd_{M1}, CCPwd_{L1}, RN_3^{Rdr}, RN_4^{Rdr}$

The reader also generates two 16-bit random numbers: $RN_1^{Rdr}$ and $RN_2^{Rdr}$. The four random numbers and the access password are used to construct $CCPwd_{M1}$ and $CCPwd_{L1}$ responses:

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \quad (1)$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \quad (2)$$

where $APWD_M$ and $APWD_L$ are the 16 most significant and 16 least significant bits of the access password, respectively. $PAD_i = PadGen\left(RN_i^{Tag}, RN_i^{Reader}\right)[APWD]$, where $PadGen(.)$ is a specially designed pad generation function. Next, two 16-bit random numbers $\left(RN_3^{Rdr}, RN_4^{Rdr}\right)$, which will be used in tag authentication, are generated and transmitted to the tag.

**Tag:** Verify $CCPwd_{M1}$ and $CCPwd_{L1}$. If both values are correct, the process continues. Otherwise, the process is aborted.

**Tag → Reader:** $RN_3^{Tag}, RN_4^{Tag}, CCPwd_{M2}, CCPwd_{L2}$

The tag also generates two new random numbers $(RN_3^{Tag}, RN_4^{Tag})$, and builds answers $CCPwd_{M2}$ and $CCPwd_{L2}$.

$$CCPwd_{M2} = APWD_M \oplus PAD_3 \quad (3)$$

$$CCPwd_{L2} = APWD_L \oplus PAD_4 \quad (4)$$

These new random numbers and answers are sent to the reader.

**Reader:** Verify $CCPwd_{M2}$ and $CCPwd_{L2}$. If both values are correct, the tag is authenticated. Otherwise an alarm is raised.

### 2.2. TRMA$^+$ scheme

In [5], Lim and Li uncovered weaknesses in Konidala and Kim's TRMA scheme. It was found that a passive attacker can recover the tag's access password by eavesdropping over a single run of the protocol and performing correlation analysis on the captured information. In [7], Konidala and Kim proposed an improved version that uses the tag's access and kill passwords. They proposed using a *PadGen* chain of length 2 (see *Section 2.3* below for details about this function). The outer *PadGen* is computed over the kill password, while the inner *PadGen* over the access password. The new scheme is essentially identical to the original TRMA scheme, but the cover-coding pad $PAD_i$ $(i = \{1, 2, 3, 4\})$ is computed differently, as follows:

$$PAD_i = PadGen(PadGen(RN_i^{Tag}, RN_i^{Reader})[APWD], RN_i^{Tag})[KPWD] \quad (5)$$

### 2.3. Pad generation function – PadGen(.)

The *PadGen* is a pad generation function that produces a 16-bit pad used to cover-code the two 16-bit access password halves ($APWD_M$ and $APWD_L$). *PadGen* takes two 16-bit input arguments and operates on a 32-bit password (*KPWD* or *APWD*) according to the input. The two input arguments are used as location indexes to retrieve individual bits from the access/kill password stored in those locations.

A detailed description of PadGen is provided below. Let the 32-bit *XPWD* (where $XPWD \in \{APWD, KPWD\}$) in binary (or Base 2) be represented as

$$XPWD = XPWD_M \| XPWD_L$$
$$XPWD_M = b_0 b_1 b_2 \ldots b_{13} b_{14} b_{15}$$
$$XPWD_L = b_{16} b_{17} b_{18} \ldots b_{29} b_{30} b_{31}$$

where each $b_i \in \{0, 1\}$. Also, let us represent the 16-bit random numbers $RN_i^{Tag}$ and $RN_i^{Rdr}$ in hexadecimal (or Base 16) representations as

$$RN_i^{Tag} = H_{i,0}^{Tag} H_{i,1}^{Tag} H_{i,2}^{Tag} H_{i,3}^{Tag}$$
$$RN_i^{Rdr} = H_{i,0}^{Rdr} H_{i,1}^{Rdr} H_{i,2}^{Rdr} H_{i,3}^{Rdr}$$

where each $H_{i,j}^{Tag}$ and $H_{i,j}^{Rdr}$ is a hexadecimal digit, i.e. $H_{i,j}^{Tag}, H_{i,j}^{Rdr} \in \boldsymbol{H}_{16} = \{0x0, 0x1, 0x2, \ldots, 0xD = 13, 0xE = 14, 0xF = 15\}$.

$PadGen(RN_i^{Tag}, RN_i^{Rdr})[XPWD]$ would then be computed as follows:

$$\begin{aligned} PadGen(RN_i^{Tag}, RN_i^{Rdr})[XPWD] = {} & b_{H_{i,0}^{Tag}} b_{H_{i,1}^{Tag}} b_{H_{i,2}^{Tag}} b_{H_{i,3}^{Tag}} \| \\ & b_{H_{i,0}^{Tag}+16} b_{H_{i,1}^{Tag}+16} b_{H_{i,2}^{Tag}+16} b_{H_{i,3}^{Tag}+16} \| \\ & b_{H_{i,0}^{Rdr}} b_{H_{i,1}^{Rdr}} b_{H_{i,2}^{Rdr}} b_{H_{i,3}^{Rdr}} \| \\ & b_{H_{i,0}^{Rdr}+16} b_{H_{i,1}^{Rdr}+16} b_{H_{i,2}^{Rdr}+16} b_{H_{i,3}^{Rdr}+16} \\ & [Base\ 2] = P_0 P_1 P_2 P_3 \quad [Base\ 16] \end{aligned}$$

for some $P_0, P_1, P_2, P_3 \in \boldsymbol{H}_{16}$.

An illustration of the *PadGen* function is shown in Fig. 1.

## 3. Access password disclosure on TRMA$^+$

In this section we describe how an attacker may succeed in recovering the 32-bit access password for a tag in the TRMA$^+$ scheme.

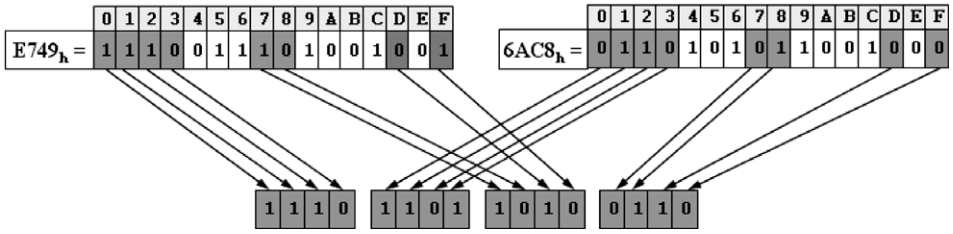


**Fig. 1.** Computing $PadGen(1234_h, 78CD_h)[E7496AC8_h]$. Note that $E749_h = 1110\ 0111\ 0100\ 1001_b$ and $6AC8_h = 0110\ 1010\ 1100\ 1000_b$.

### 3.1. Access password attack (LSB)

The attack is outlined in the following diagram. Details follow below:

(1) $\mathbf{T} \rightarrow \mathbf{R} : \{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2) $\vdots$
(3) $\mathbf{A} \rightarrow \mathbf{R} : \{EPC, RN_1^{Tag'}, RN_2^{Tag'}\}$
(4) $\mathbf{R} \rightarrow \mathbf{A} : \{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$

T, tag; R, reader; A, attacker

*Scenario*: An adversary eavesdrops an authentication session between a genuine reader and a genuine tag to obtain a valid EPC. This tag then becomes the target of the attack. With the EPC obtained, the adversary effects an active attack by masquerading as the target tag and participating in the TRMA$^+$ protocol with a genuine reader. The adversary sends the message $\left\{EPC, RN_1^{Tag'}, RN_2^{Tag'}\right\}$ to the reader such that all the hexadecimal digits in each of $RN_1^{Tag'}$ and $RN_2^{Tag'}$ have the same value:

$$RN_i^{Tag'} = RRRR_h \quad [Base\ 16] \tag{6}$$

where $R \in \boldsymbol{H}_{16}$ and $RN_1^{Tag'}$ may or may not be equal to $RN_2^{Tag'}$. Next, the adversary receives the response provided by the reader $\left\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\right\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \tag{7}$$
$$CCPwd_{L1} = APWD_L \oplus PAD_2 \tag{8}$$

and for $i \in \{1, 2\}$,

$$PAD_i = PadGen(PadGen(RN_i^{Tag'}, RN_i^{Rdr})[APWD], RN_i^{Tag'})[KPWD] \tag{9}$$

Let $PadGen\left(RN_i^{Tag'}, RN_i^{Rdr}\right)[APWD] = V_0V_1V_2V_3$ for some hexadecimal digits $V_0, V_1, V_2, V_3 \in \boldsymbol{H}_{16}$. Substituting this and (6) into (9), we have

$$\begin{aligned} PAD_i &= PadGen(V_0V_1V_2V_3, RRRR)[KPWD] \quad [Base\ 16] \\ &= k_{V_0}k_{V_1}k_{V_2}k_{V_3} \| k_{V_0+16}k_{V_1+16}k_{V_2+16}k_{V_3+16} \| k_Rk_Rk_Rk_R \| \\ &\quad k_{R+16}k_{R+16}k_{R+16}k_{R+16} \quad [Base\ 2] \\ &= P_0P_1P_2P_3 \quad [Base\ 16] \end{aligned}$$

where each $k_j$ is the $j^{th}$ bit in the kill password. We observe that all the bits in each of the hexadecimal digits $P_2$ and $P_3$ are the same, i.e. $P_2, P_3 \in \{0000_b = 0_h, 1111_b = F_h\}$. This leads to $P_2P_3 \in \{00_h, 0F_h, F0_h, FF_h\}$. Assuming that $P_2P_3$ takes each value with equal probability, the adversary can then use this to obtain the 8 least significant bits of $APWD_L$ and $APWD_M$ by computing the following:

$$APWD_{M_{[8\ldots15]}} = \begin{cases} CCPwd_{M1_{[8\ldots15]}} \oplus 0x00 & \text{with } p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0x0F & \text{with } p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0xF0 & \text{with } p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0xFF & \text{with } p = 2^{-2} \end{cases} \tag{10}$$

$$APWD_{L_{[8\ldots15]}} = \begin{cases} CCPwd_{L1_{[8\ldots15]}} \oplus 0x00 & \text{with } p = 2^{-2} \\ CCPwd_{L1_{[8\ldots15]}} \oplus 0x0F & \text{with } p = 2^{-2} \\ CCPwd_{L1_{[8\ldots15]}} \oplus 0xF0 & \text{with } p = 2^{-2} \\ CCPwd_{L1_{[8\ldots15]}} \oplus 0xFF & \text{with } p = 2^{-2} \end{cases} \tag{11}$$

Summarizing, the adversary can obtain the 8 least significant bits of $APW_M$ and $APW_L$ with probability $2^{-2}$ for each. The attack is more powerful if the random numbers are such that $RN_1^{Tag'} = RN_2^{Tag'}$. Under this condition, an adversary may also extract the following 16-bits of the access password with probability $2^{-2}$:

$$\begin{aligned} &APWD_{M_{[8\ldots15]}} \| APWD_{L_{[8\ldots15]}} \\ &= \begin{cases} CCPwd_{M1_{[8\ldots15]}} \oplus 0x00 \| CCPwd_{L1_{[8\ldots15]}} \oplus 0x00, & p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0x0F \| CCPwd_{L1_{[8\ldots15]}} \oplus 0x0F, & p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0xF0 \| CCPwd_{L1_{[8\ldots15]}} \oplus 0xF0, & p = 2^{-2} \\ CCPwd_{M1_{[8\ldots15]}} \oplus 0xFF \| CCPwd_{L1_{[8\ldots15]}} \oplus 0xFF, & p = 2^{-2} \end{cases} \end{aligned} \tag{12}$$

Hence, an active attacker can gather vast amounts of information about the tag's access password in a single run of the TRMA$^+$ protocol.

### 3.2. Access password attack (MSB)

The attack (man-in-the-middle) is outlined in the following diagram. Details follow below:

**Session A**:
(1) $\mathbf{T} \rightarrow \mathbf{A}$: $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2a) $\mathbf{A} \rightarrow \mathbf{R}$: $\{EPC, RN, RN\}$
(2b) $\mathbf{R} \rightarrow \mathbf{A}$:
$\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$

**Session B:**
(3a) $\mathbf{A} \rightarrow \mathbf{R}$: $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(3b) $\mathbf{R} \rightarrow \mathbf{A}$:
$\{CCPwd'_{M1}, CCPwd'_{L1}, RN_1^{Rdr'}, RN_2^{Rdr'}, RN_3^{Rdr'}, RN_4^{Rdr'}\}$
(4) $\mathbf{A} \rightarrow \mathbf{T}$: $\{CCPwd'_{M1}, CCPwd'_{L1}, RN_1^{Rdr'}, RN_2^{Rdr'}, RN, RN\}$
(5) $\mathbf{T} \rightarrow \mathbf{A}$: $\{RN_3^{Tag}, RN_4^{Tag}, CCPwd_{M2}, CCPwd_{L2}\}$

T, tag; R, reader; A, attacker

*Scenario*: An adversary intercepts and alters the content of the message sent by a genuine tag (Session A). The random numbers picked up by the adversary are then set to *RN* before being forwarded to the reader. Specifically, the random number *RN* must satisfy the following equation:

$$RN = RRRR_h \quad [Base\ 16] \tag{13}$$

where $R \in \boldsymbol{H}_{16}$. The adversary receives the response provided by the legitimate reader: $\left\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\right\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \quad (14)$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \quad (15)$$

and for $i \in \{1,2\}$,

$$PAD_i = PadGen(PadGen(RN, RN_i^{Rdr})[APWD], RN)[KPWD] \quad (16)$$

In a different, parallel authentication session (Session B), the adversary forwards the initial message sent by the tag $\left\{EPC, RN_1^{Tag}, RN_2^{Tag}\right\}$ to the legitimate reader. The reader's response $\left\{CCPwd'_{M1}, CCPwd'_{L1}, RN_1^{Rdr'}, RN_2^{Rdr'}, RN_3^{Rdr'}, RN_4^{Rdr'}\right\}$ is received by the adversary, who then sets the random numbers $RN_3^{Rdr'}$ and $RN_4^{Rdr'}$ to $RN$. (Note that these two random numbers will be used by the tag to compute its response to a reader). The modified message is then forwarded to the genuine tag, which responds by sending the message: $\left\{CCPwd_{M2}, CCPwd_{L2}, RN_3^{Tag}, RN_4^{Tag}\right\}$, where

$$CCPwd_{M2} = APWD_M \oplus PAD_3 \quad (17)$$

$$CCPwd_{L2} = APWD_L \oplus PAD_4 \quad (18)$$

and for $i \in \{3,4\}$,

$$PAD_i = PadGen\left(PadGen\left(RN_i^{Tag}, RN\right)[APWD], RN_i^{Tag}\right)[KPWD] \quad (19)$$

In this attack scenario, the adversary can derive the following:

**1. Information from the computation of** $\mathrm{PAD}_{i\in\{1,2\}}$

$$\begin{aligned} &PadGen(RN, RN_i^{Rdr})[APWD] \\ &= PadGen(RRRR, H_{i,0}^{Rdr}H_{i,1}^{Rdr}H_{i,2}^{Rdr}H_{i,3}^{Rdr})[APWD] \quad [Base\ 16] \\ &= a_R a_R a_R a_R \| a_{R+16} a_{R+16} a_{R+16} a_{R+16} \| a_{H_{i,0}^{Rdr}} a_{H_{i,1}^{Rdr}} a_{H_{i,2}^{Rdr}} a_{H_{i,3}^{Rdr}} \| \\ &a_{H_{i,0}^{Rdr}+16} a_{H_{i,1}^{Rdr}+16} a_{H_{i,2}^{Rdr}+16} a_{H_{i,3}^{Rdr}+16} \quad [Base\ 2] \\ &= V_0V_1V_2V_3 \quad [Base\ 16] \end{aligned} \quad (20)$$

where we observe that all four bits in each of $V_0$ and $V_1$ have the same value, i.e. $V_0, V_1 \in \{0_h, \mathrm{F}_h\}$ or $V_0V_1 \in \{00_h, 0\mathrm{F}_h, \mathrm{F}0_h, \mathrm{FF}_h\}$ (as in the previous attack on the LSB described in *Section 3.1*). Then,

$$\begin{aligned} PAD_{i\in\{1,2\}} &= PadGen(PadGen(RN, RN_1^{Rdr})[APWD], RN)[KPWD] \\ &= PadGen(V_0V_1V_2V_3, RRRR)[KPWD] \quad [Base\ 16] \\ &= k_{V_0}k_{V_1}k_{V_2}k_{V_3} \| k_{V_0+16}k_{V_1+16}k_{V_2+16}k_{V_3+16} \| k_R k_R k_R k_R \| \\ &\quad k_{R+16}k_{R+16}k_{R+16}k_{R+16} \quad [Base\ 2] \end{aligned} \quad (21)$$

Assuming that the values of $V_0$ and $V_1$ are taken randomly from the set $\{0_h, \mathrm{F}_h\}$, then they will be equal half of the time, i.e. with probability 0.5. If $V_0 = V_1$, then $k_{V_0} = k_{V_1}$. On the other hand, if $V_0 \neq V_1$, then assuming that the bits in the kill password are perfectly random, we will have $k_{V_0} = k_{V_1}$ with probability 0.5. Hence,

$$Prob(k_{V_0} = k_{V_1}) = (0.5)(1) + (0.5)(0.5) = 0.75 \quad (22)$$

Similarly, $Prob(k_{V_0+16} = k_{V_1+16}) = 0.75$.

**2. Information from the computation of** $\mathrm{PAD}_{i\in\{3,4\}}$

$$\begin{aligned} &PadGen(RN_i^{Tag}, RN)[APWD] \\ &= PadGen(H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag}, RRRR)[APWD] \quad [Base\ 16] \\ &= a_{H_{i,0}^{Tag}} a_{H_{i,1}^{Tag}} a_{H_{i,2}^{Tag}} a_{H_{i,3}^{Tag}} \| a_{H_{i,0}^{Tag}+16} a_{H_{i,1}^{Tag}+16} a_{H_{i,2}^{Tag}+16} a_{H_{i,3}^{Tag}+16} \| \\ &\quad a_R a_R a_R a_R \| a_{R+16} a_{R+16} a_{R+16} a_{R+16} \quad [Base\ 2] \\ &= S_0S_1S_2S_3 \quad [Base\ 16] \end{aligned} \quad (23)$$

where $S_2, S_3 \in \{0_h, \mathrm{F}_h\}$ or $S_2S_3 \in \{00_h, 0\mathrm{F}_h, \mathrm{F}0_h, \mathrm{FF}_h\}$. Furthermore, we note that $V_0 = S_2$ and $V_1 = S_3$. Next, we derive

$$\begin{aligned} PAD_{i\in\{3,4\}} &= PadGen(PadGen(RN_i^{Tag}, RN)[APWD], RN_i^{Tag})[KPWD] \\ &= PadGen(S_0S_1S_2S_3, H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag})[KPWD] \quad [Base\ 16] \\ &= k_{S_0}k_{S_1}k_{S_2}k_{S_3} \| k_{S_0+16}k_{S_1+16}k_{S_2+16}k_{S_3+16} \| k_{H_{i,0}^{Tag}} k_{H_{i,1}^{Tag}} k_{H_{i,2}^{Tag}} k_{H_{i,3}^{Tag}} \| \\ &\quad k_{H_{i,0}^{Tag}+16} k_{H_{i,1}^{Tag}+16} k_{H_{i,2}^{Tag}+16} k_{H_{i,3}^{Tag}+16} \quad [Base\ 2] \end{aligned} \quad (24)$$

As in the earlier case for the computation of $PAD_{i\in\{1,2\}}$, assuming that $S_2 = S_3$ half of the time, we have $Prob(k_{S_2} = k_{S_3}) = 0.75$ and $Prob(k_{S_2+16} = k_{S_3+16}) = 0.75$.

**3. Combining both sets of information**

Since $V_0 = S_2$ and $V_1 = S_3$, we then have

$$k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}, \quad p = 0.75$$

$$k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3}, \quad p = 0.25$$

and

$$k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}, \quad p = 0.75$$
$$k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} = k_{S_3+16}, \quad p = 0.25$$

However, instead of considering these two sets of relations separately, we combine them to give four possible cases, and their corresponding probabilities can be computed as follows:

- **Case 1:** $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$. The two relations will always hold if $V_0 = V_1$ (which also implies $S_2 = S_3$). When $V_0 \neq V_1$ (and $S_2 \neq S_3$), the probability that $k_{V_0} = k_{V_1}$ (and $k_{S_2} = k_{S_3}$) is 0.5. Similarly, the probability that $k_{V_0+16} = k_{V_1+16}$ (and $k_{S_2+16} = k_{S_3+16}$) is also 0.5. Hence, the probability that this case will occur is (0.5)(1) + (0.5)(0.5)(0.5)=0.625.
- **Case 2:** $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} = k_{S_3+16}$. This case will only occur when $V_0 \neq V_1$ (and $S_2 \neq S_3$). In such a situation, the probability that $k_{V_0} = k_{V_1}$ (and $k_{S_2} = k_{S_3}$) is 0.5, and the probability that $k_{V_0+16} \neq k_{V_1+16}$ (and $k_{S_2+16} \neq k_{S_3+16}$) is 0.5. Hence, the probability that the two relations will hold is (0.5)(0.5)(0.5)=0.125.
- **Case 3:** $k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$. This case is similar to Case 2 and occurs when $V_0 \neq V_1$ $(S_2 \neq S_3), k_{V_0} \neq k_{V_1} (k_{S_2} \neq k_{S_3})$ but $k_{V_0+16} = k_{V_1+16} (k_{S_2+16} = k_{S_3+16})$. The resulting probability for this case is (0.5)(0.5)(0.5)=0.125.
- **Case 4:** $k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} = k_{S_3+16}$. This case is also similar to Case 2 and occurs when $V_0 \neq V_1$ $(S_2 \neq S_3), k_{V_0} \neq k_{V_1} (k_{S_2} \neq k_{S_3})$ and $k_{V_0+16} \neq k_{V_1+16} (k_{S_2+16} \neq k_{S_3+16})$. It yields a probability of (0.5)(0.5)(0.5)=0.125.

Based on this information, the 8 most significant bits of $APWD_M$ and $APWD_L$ can be given by

$$APWD_{M_{[0\ldots7]}} \| APWD_{L_{[0\ldots7]}} = A \oplus mask \| B \oplus mask \quad (25)$$

where

$$A = (CCPwd_{M1_{[0\ldots7]}} \wedge 0\mathrm{xCC}) \vee (CCPwd_{M2_{[0\ldots7]}} \wedge 0\mathrm{x33}) \quad (26)$$

$$B = (CCPwd_{L1_{[0\ldots7]}} \wedge 0\mathrm{xCC}) \vee (CCPwd_{L2_{[0\ldots7]}} \wedge 0\mathrm{x33}) \quad (27)$$

and $\wedge$ denotes the bitwise logical AND operation, $\vee$ denotes the bitwise logical OR operation. The *mask* in (25) can take a number of probable values depending on whether Case 1, 2, 3 or 4 holds:

- If **Case 1** holds, i.e. $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$, then $mask \in \{0\mathrm{x}00, 0\mathrm{x}0\mathrm{F}, 0\mathrm{x}\mathrm{F}0, 0\mathrm{x}\mathrm{FF}\}$. In this case, if the adversary were to select a mask from the specified set of values, the probability of a successful attack to recover all 16 bits of the access password would be

$$Prob(\text{successful recovery of all bits in } APWD_{M_{[0\ldots7]}} \| APWD_{L_{[0\ldots7]}}) = 0.625 \times 1/4 = 0.15625 \quad (28)$$

- If **Case 2** holds, then $mask \in \{0x05, 0x0A, 0xF5, 0xFA\}$ and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.
- If **Case 3** holds, then $mask \in \{0x50, 0x5F, 0xA0, 0xAF\}$ and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.
- If **Case 4** holds, then $mask \in \{0x55, 0x5A, 0xA5, 0xAA\}$ and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.

In summary, with Eqs. (25)–(27), the probability of a successful attack for any selected mask would be given by

$$Prob(\text{successful recovery of all bits in} APWD_{M_{[0\ldots7]}} \| APWD_{L_{[0\ldots7]}}) = \begin{cases} \frac{5}{2^5} = 0.15625 & \text{if } mask \in \{0x00, 0x0F, 0xF0, 0xFF\} \\ \frac{1}{2^5} = 0.03125 & \text{if } mask \in \{0x05, 0x0A, 0x50, 0x55, \\ & \quad 0x5A, 0x5F, 0xA0, 0xA5, 0xAA, \\ & \quad 0xAF\ 0xF5, 0xFA\} \end{cases} \quad (29)$$

Hence, in order to maximize the probability of success of an attack, the adversary should select *mask* from the set $\{0x00, 0x0F, 0xF0, 0xFF\}$. In any case, this attack results in 16 possible values for the most significant bits of $APWD_M$ and $APWD_L$. Together with the four possible values for the least significant bits of $APWD_M$ and $APWD_L$ obtained from the earlier attack, the adversary can narrow down the possible values for the access password from $2^{32}$ to $16 \times 4 = 2^6$, which is a tremendous reduction.

## 4. Kill password disclosure on TRMA+

In this section, we describe how an attacker can succeed in recovering the 32-bit kill password for a tag in the TRMA+ scheme. We present two different approaches towards this objective. First, we assume that the attacker has no information (i.e. APWD, KPWD) about the target tag (see Section 4.1). Second, we assume that the attacker knows the 8 least significant bits of the access password (see Section 4.2).

### 4.1. Correlation attack

In this attack, the adversary first performs passive eavesdropping on the messages exchanged during an authentication session. He will obtain all the $CCPwd_{Mi}, CCPwd_{Li}$ (for $i = 1, 2$) and $RN_i^{Tag}, RN_i^{Rdr}$ (for $i = 1, 2, 3, 4$) since all these one are transmitted unencrypted.

We examine the relationship between the eavesdropped messages $CCPwd_{Mi}, CCPwd_{Li}$ (for $i = 1, 2$) and $RN_i^{Tag}, RN_i^{Rdr}$ (for $i = 1, 2, 3, 4$) and the tag secrets (*APWD* and *KPWD*). Let $[Z]_b$ denote the $b^{th}$ bit value of $Z$ and $\langle Z \rangle_x$ denote the $x^{th}$ hexadecimal value of $Z$ with the convention that the $0^{th}$ value is the most significant value. From Eqs. (1) and (5), we can represent the bits of $CCPwd_{M1}$ as follows:

$$[CCPwd_{M1}]_0 = [APWD_M]_0 \oplus [KPWD_M]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_0}$$

$$[CCPwd_{M1}]_1 = [APWD_M]_1 \oplus [KPWD_M]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_1}$$

$$[CCPwd_{M1}]_2 = [APWD_M]_2 \oplus [KPWD_M]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_2}$$

$$[CCPwd_{M1}]_3 = [APWD_M]_3 \oplus [KPWD_M]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_3}$$

$$[CCPwd_{M1}]_4 = [APWD_M]_4 \oplus [KPWD_L]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_0}$$

$$[CCPwd_{M1}]_5 = [APWD_M]_5 \oplus [KPWD_L]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_1}$$

$$[CCPwd_{M1}]_6 = [APWD_M]_6 \oplus [KPWD_L]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_2}$$

$$[CCPwd_{M1}]_7 = [APWD_M]_7 \oplus [KPWD_L]_{\langle PadGen(RN_1^{Tag}, RN_1^{Rdr})[APWD] \rangle_3}$$

$$[CCPwd_{M1}]_8 = [APWD_M]_8 \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_0}$$

$$[CCPwd_{M1}]_9 = [APWD_M]_9 \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_1}$$

$$[CCPwd_{M1}]_A = [APWD_M]_A \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_2}$$

$$[CCPwd_{M1}]_B = [APWD_M]_B \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_3}$$

$$[CCPwd_{M1}]_C = [APWD_M]_C \oplus [KPWD_L]_{\langle RN_1^{Tag} \rangle_0}$$

$$[CCPwd_{M1}]_D = [APWD_M]_D \oplus [KPWD_L]_{\langle RN_1^{Tag} \rangle_1}$$

$$[CCPwd_{M1}]_E = [APWD_M]_E \oplus [KPWD_L]_{\langle RN_1^{Tag} \rangle_2}$$

$$[CCPwd_{M1}]_F = [APWD_M]_F \oplus [KPWD_L]_{\langle RN_1^{Tag} \rangle_3}$$

A similar representation can be obtained for $CCPwd_{M2}$ (note that for $CCPwd_{L1}$ and $CCPwd_{L2}$, the first operand would be $APWD_L$, instead of $APWD_M$). We find that by considering the eight least significant bits (bits $8_h$ to $F_h$) of $CCPwd_{M1}$ and the corresponding bits of $CCPwd_{M2}$, an adversary would be able to obtain bit correlations in $KPWD_M$ and $KPWD_L$ if $RN_1^{Tag}$ and $RN_3^{Tag}$ are known. For example, consider $[CCPwd_{M1}]_8 = [APWD_M]_8 \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_0}$ and $[CCPwd_{M2}]_8 = [APWD_M]_8 \oplus [KPWD_M]_{\langle RN_3^{Tag} \rangle_0}$. Performing XOR over these two bits will cancel out $[APWD_M]_8$ and give us a correlation between $[KPWD_M]_{\langle RN_1^{Tag} \rangle_0}$ and $[KPWD_M]_{\langle RN_3^{Tag} \rangle_0}$:

$$\begin{aligned} [CCPwd_{M1}]_8 \oplus [CCPwd_{M2}]_8 &= \left([APWD_M]_8 \oplus [KPWD_M]_{\langle RN_1^{Tag} \rangle_0}\right) \\ &\quad \oplus \left([APWD_M]_8 \oplus [KPWD_M]_{\langle RN_3^{Tag} \rangle_0}\right) \\ &= [KPWD_M]_{\langle RN_1^{Tag} \rangle_0} \oplus [KPWD_M]_{\langle RN_3^{Tag} \rangle_0} \end{aligned} \quad (30)$$

If $[CCPwd_{M1}]_8 \oplus [CCPwd_{M2}]_8 = 0$, we can deduce that $[KPWD_M]_{\langle RN_1^{Tag} \rangle_0} = [KPWD_M]_{\langle RN_3^{Tag} \rangle_0}$. On the other hand, if $[CCPwd_{M1}]_8 \oplus [CCPwd_{M2}]_8 = 1$, then $[KPWD_M]_{\langle RN_1^{Tag} \rangle_0} = \overline{[KPWD_M]_{\langle RN_3^{Tag} \rangle_0}}$ (where we take $\overline{X}$ to denote the negation of $X$). Together with the other seven least significant bits of $CCPwd_{M1}$ and $CCPwd_{M2}$, we can derive four pairs of correlations in $KPWD_M$ and another four pairs of correlations in $KPWD_L$. The same applies to $CCPwd_{L1}$ and $CCPwd_{L2}$. Hence, every authentication session can reveal up to eight pairs of correlations each for $KPWD_M$ and $KPWD_L$.

By eavesdropping over a sufficient number of authentication sessions involving the target tag, the adversary would then be able to correlate all the bits in $KPWD_M$ and $KPWD_L$. With a full set of correlations, the adversary would arrive at two possible values (each a negation of the other) for $KPWD_M$, as well as $KPWD_L$. In other words, the adversary can obtain the kill password ($KPW_M$ and $KPW_L$) with probability $2^{-2}$.

We performed a simulation to investigate the average number of authentication sessions required for an adversary to obtain the full set of correlations involving $KPWD_M$ and $KPWD_L$. In the experiment, we simulated an attack scenario in which an adversary eavesdrops over multiple authentication sessions (up to a maximum of 20 sessions) until the maximal set of correlations is collected for $KPWD_M$ and $KPWD_L$. The number of eavesdropped sessions required to accumulate the full set of correlations was then recorded. This was repeated over 20,000 times. In the experiment, the mean number of sessions required to accumulate the full set of correlations was 4.075 and the mode was 4.

### 4.2. Cover-code attack

In Section 3.1, we showed how an attacker is able to obtain the 8 least significant bits of $APWD_M$ and $APWD_L$, each with probability $2^{-2}$. This advantage can be employed by an adversary to recover the 32 bits of the kill password with the same probability. The attack is described below.

Given that the adversary knows the access password of the target tag, the pads used to cover-code the MSB and LSB of the access password can be obtained as follows:

$$PAD1_{[8\ldots15]} = CCPwd_{M1_{[8\ldots15]}} \oplus APWD_{M_{[8\ldots15]}} \quad (31)$$

$$PAD2_{[8\ldots15]} = CCPwd_{L1_{[8\ldots15]}} \oplus APWD_{L_{[8\ldots15]}} \quad (32)$$

$$PAD3_{[8\ldots15]} = CCPwd_{M2_{[8\ldots15]}} \oplus APWD_{M_{[8\ldots15]}} \quad (33)$$

$$PAD4_{[8\ldots15]} = CCPwd_{L2_{[8\ldots15]}} \oplus APWD_{L_{[8\ldots15]}} \quad (34)$$

where the 8 bits in each pad $PAD_i$ are bits selected from different memory locations in the kill password:

$$PAD_i = PadGen\left(----, RN_i^{Tag}\right)[KPWD]$$
$$= PadGen\left(----, H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag}\right)[KPWD] \quad [Base\ 16]$$

Hence,

$$PAD_{i_{[8\ldots15]}} = k_{H_{i,0}^{Tag}}k_{H_{i,1}^{Tag}}k_{H_{i,2}^{Tag}}k_{H_{i,3}^{Tag}} \| k_{H_{i,0}^{Tag}+16}k_{H_{i,1}^{Tag}+16}k_{H_{i,2}^{Tag}+16}k_{H_{i,3}^{Tag}+16} \quad [Base\ 2]$$

So, we have the following equations relating a bit in the kill password to a bit in each $PAD_i (i \in \{1,2,3,4\})$:

$$k_{H_{i,0}^{Tag}} = PAD_{i_{[8]}}, \quad k_{H_{i,0}^{Tag}+16} = PAD_{i_{[12]}}$$
$$k_{H_{i,1}^{Tag}} = PAD_{i_{[9]}}, \quad k_{H_{i,1}^{Tag}+16} = PAD_{i_{[13]}}$$
$$k_{H_{i,2}^{Tag}} = PAD_{i_{[10]}}, \quad k_{H_{i,2}^{Tag}+16} = PAD_{i_{[14]}}$$
$$k_{H_{i,3}^{Tag}} = PAD_{i_{[11]}}, \quad k_{H_{i,3}^{Tag}+16} = PAD_{i_{[15]}}$$

where each bit $PAD_i[n]$ can be computed using one of the equations (31)–(33) or (34). For example, $k_{H_{1,1}^{Tag}} = PAD_1[9] = CCPwd_{M1}[9] \oplus APWD_M[9]$ and $k_{H_{4,0}^{Tag}+16} = PAD_4[12] = CCPwd_{L2}[12] \oplus APWD_L[12]$. Hence, when an adversary has obtained the LSB of the access password (see *Section 3.1*), he is able to obtain the bits in the kill password. For recovery of the entire kill password, there are two possible approaches:

*Passive attacker:* In this case, an adversary eavesdrops over multiple sessions of the protocol in order to obtain all 32 bits of the kill password. In our experiments, which simulated 20,000 executions of the attack, the average number of sessions required to obtain the full kill password was 4. Fig. 2(a) and (c) shows

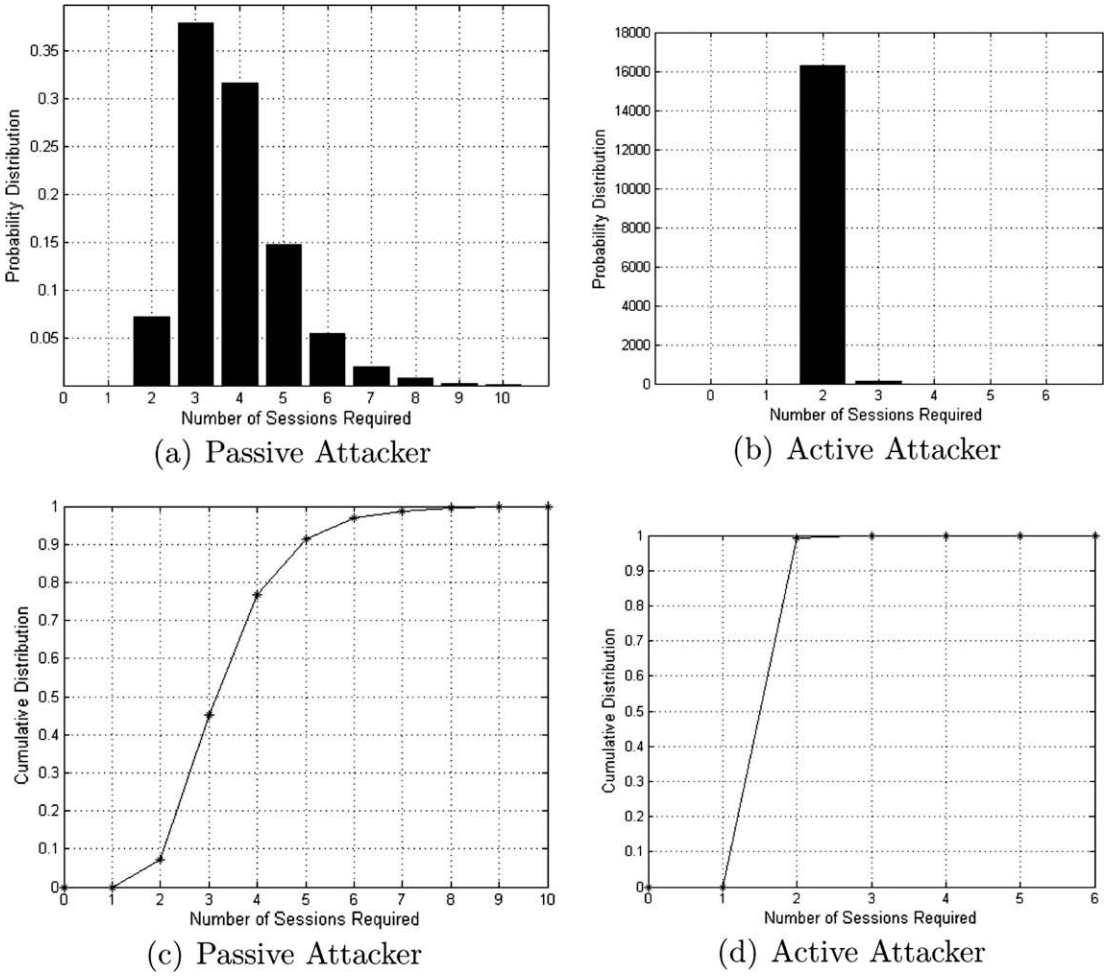


**Fig. 2.** Probability and cumulative distributions for the number of sessions required for a successful attack.

the probability and cumulative distributions for the number of sessions required.

*Active attacker:* In the active attack, the adversary modifies and manipulates the random numbers $RN_1^{Tag}$ and $RN_2^{Tag}$ to lead the legitimate reader to select bits in the kill password in such a way that avoids those bit locations where the value of the bit is already known. Hence, the number of sessions required to obtain the all 32 bits of the kill password is reduced. The results of our experiments show that the average number of attack sessions required to obtain the full password is 2, i.e. a 50% reduction from the passive attack. Fig. 2(b) and (d) shows the results.

## 5. Conclusion

In this paper, we have shown the existence of significant security flaws in a lightweight mutual authentication scheme[1] conforming to the EPC-C1G2 standard. The protocol uses the access and kill passwords defined in the specification, which are shared between legitimate entities (tags and readers). The authors suggested the use of a pad generation (*see Section 2.3*) function to protect both passwords. This function, however, is not secure enough, as the cover-codes generated depend on random numbers selected by the tag/reader. We have found that after simple computations an attacker can acquire the access and kill passwords with high probability.

Specifically, the most and least significant eight bits of the access password can be obtained with a probability of $2^{-5}$ and $2^{-2}$, respectively. The kill password can be acquired over two different approximations. First, we assumed the weakest attack scenario in which the attacker has no information about the target tag and only eavesdrops on the channel. Under these conditions, once several authentication sessions are eavesdropped, the attacker can derive four possible values of the kill password. Secondly, we assumed that the attacker has previously performed the access password attack (see Section 3.1) and knows the least significant bits of the access password. As in the above case, all 32 bits of the access password can be derived with a probability of $2^{-2}$. The efficiency of this attack depends on whether a passive or active attacker is assumed. A passive attacker has to eavesdrop an average of 4 protocol rounds, but this number is reduced to 2 when the attacker can modify and manipulate the exchanged messages.

These attacks have serious implications. First, knowledge of the access password allows complete access to the tag's memory. When the access password is received by a tag, it transitions from an open to a secured state. In this state, the attacker can execute one of the following commands: *Read*, *Write*, *BlockWrite*, and *BlockErase* (see EPC-C1G2 for more details [10]). Second, the attacker can leave inoperative the target tag, frustrating its intended identification, when the kill password is known. If this attack is launched on a large scale, it can lead to a denial information attack. With this in mind, we recommend that the kill password only be used to allow an authorized user to kill the tag, and not for other purposes such as access password protection.

In summary, the EPC-C1G2 standard has important security flaws (e.g. access/kill password disclosure) [11]. To date, we have found that most authentication protocols designed in compliance with EPC-C1G2 are vulnerable to attack, even under a weak security model (as shown in [3–5]). The designing of new authentication protocols conforming to the standard and providing an adequate security level is therefore an exciting challenge.

[1] This paper was presented at the RFIDSec'07 conference, considered one of the most important annual events in RFID security.


## Acknowledgement

We are grateful to Tong-Lee Lim for helpful comments and invaluable help.


## Appendix A

To clarify to the reader how the access password and kill password attacks could be accomplished, we illustrate these attacks below, using a simple example. We assume the following values for the target tag: *EPC*, *KPWD*(0x123 F80*C*0), and *APWD*(0x45AB 7FDC).

## Appendix B. Example of access password attack (LSB)

An attacker must eavesdrop on a legitimate session (reader-tag) before the active attack can be carried out. An example of the attack is shown below.

**Eavesdropped session**

The attacker acquires the *EPC* of the target tag.

**Active attack**

*Attacker* $\rightarrow$ *Reader* : $\{EPC, 0x1111, 0x1111\}$

Reader $\rightarrow$ Attacker: {0xB5AB, 0xAFDC, 0x56A1, 0x3297, 0xAB42, 0xD65C}

The adversary extracts four possible values for access password:

$$APWD_{M_{[8\ldots15]}} \| APWD_{L_{[8\ldots15]}} = \begin{cases} 0xAB \oplus 0x00 = \mathbf{0xAB} \| 0xDC \oplus 0x00 = \mathbf{0xDC}, & p = 2^{-2} \\ 0xAB \oplus 0x0F = 0xA4 \| 0xDC \oplus 0x0F = 0xD3, & p = 2^{-2} \\ 0xAB \oplus 0xF0 = 0x5B \| 0xDC \oplus 0xF0 = 0x2C, & p = 2^{-2} \\ 0x54 \oplus 0xFF = 0x54 \| 0xDC \oplus 0xFF = 0x23, & p = 2^{-2} \end{cases} \quad (35)$$

## Appendix C. Example of access password attack (MSB)

An active attacker must intercept, modify and re-forward the messages sent between a legitimate tag and reader, over two sessions, in order to complete the attack. We present an example of this attack below.

**Session A**

(1) **Tag** $\rightarrow$ **Attacker** : $\{EPC, 0xAB43, 0x769F\}$
(2a) **Attacker** $\rightarrow$ **Reader** : $\{EPC, 0x5555, 0x555\}$
(2b) **Reader** $\rightarrow$ **Attacker** :
$\{0x95AB, 0x8DDC, 0x347B, 0xB391, 0x789A, 0x6733\}$

**Session B**

(3a) **Attacker** $\rightarrow$ **Reader** : $\{EPC, 0xAB43, 0x769F\}$
(3b) **Reader** $\rightarrow$ **Attacker** :
$\{0x7D7B, 0x075C0xC387, 0x6712, 0x456A, 0xFAB1\}$
(4) **Attacker** $\rightarrow$ **Tag** :
$\{0x7D7B, 0x075C0xC387, 0x6712, 0x5555, 0x5555\}$
(5) **Tag** $\rightarrow$ **Attacker** :
$\{0x3277, 0xC761, 0xB5AB, 0x8F7C\}$

The adversary extracts 16 possible values for access password:

$$A = (0x95 \wedge 0xCC) \vee (0xB5 \wedge 0x33) = 0xB5 \quad (36)$$

$$B = (0x8D \wedge 0xCC) \vee (0x8F \wedge 0x33) = 0x8F \quad (37)$$

$$APWD_{M_{[0\ldots7]}} \| APWD_{L_{[0\ldots7]}} = \begin{cases} 0xB5 \oplus 0x00 = 0xB5 \| 0x8F \oplus 0x00 = 0x8F, & p = 0.15625 \\ 0xB5 \oplus 0x0F = 0xBA \| 0x8F \oplus 0x0F = 0x80, & p = 0.15625 \\ 0xB5 \oplus 0xF0 = 0x45 \| 0x8F \oplus 0xF0 = 0x7F, & p = 0.15625 \\ 0xB5 \oplus 0xFF = 0x4A \| 0x8F \oplus 0xFF = 0x70, & p = 0.15625 \\ 0xB5 \oplus 0x05 = 0xB0 \| 0x8F \oplus 0x05 = 0x8A, & p = 0.03125 \\ 0xB5 \oplus 0x0A = 0xBF \| 0x8F \oplus 0x0A = 0x85, & p = 0.03125 \\ 0xB5 \oplus 0x50 = 0xE5 \| 0x8F \oplus 0x50 = 0xDF, & p = 0.03125 \\ 0xB5 \oplus 0x55 = 0xE0 \| 0x8F \oplus 0x55 = 0xDA, & p = 0.03125 \\ 0xB5 \oplus 0x5A = 0xEF \| 0x8F \oplus 0x5A = 0xD5, & p = 0.03125 \\ 0xB5 \oplus 0x5F = 0xEA \| 0x8F \oplus 0x5F = 0xD0, & p = 0.03125 \\ 0xB5 \oplus 0xA0 = 0x15 \| 0x8F \oplus 0xA0 = 0x2F, & p = 0.03125 \\ 0xB5 \oplus 0xA5 = 0x10 \| 0x8F \oplus 0xA5 = 0x2A, & p = 0.03125 \\ 0xB5 \oplus 0xAA = 0x1F \| 0x8F \oplus 0xAA = 0x25, & p = 0.03125 \\ 0xB5 \oplus 0xAF = 0x1A \| 0x8F \oplus 0xAF = 0x20, & p = 0.03125 \\ 0xB5 \oplus 0xF5 = 0x40 \| 0x8F \oplus 0xF5 = 0x7A, & p = 0.03125 \\ 0xB5 \oplus 0xFA = 0x4F \| 0x8F \oplus 0xFA = 0x75, & p = 0.03125 \end{cases} \quad (38)$$

## Appendix D. An example of Kill password attack

To effect the attack, the adversary performs passive eavesdropping over multiple authentication sessions involving the target tag, and computes the cumulative correlations obtained after each session, until the maximal set of correlations is obtained. An example of the attack follows.

### Eavesdropped Session 1:

$RN_1^{Tag} = 0xA44,\ RN_2^{Tag} = 0x520A,\ RN_3^{Tag} = 0x315F,$
$RN_4^{Tag} = 0xED07$

(To keep things simple, only the random numbers generated by the tag are shown.)

Correlations obtained by considering $RN_1^{Tag}$ and $RN_3^{Tag}$, and XORing the relevant bits in $CCPwd_{M1}$ and $CCPwd_{M2}$:

$$[KPWD_M]_0 = \overline{[KPWD_M]_3}, \quad [KPWD_M]_A = \overline{[KPWD_M]_1},$$
$$[KPWD_M]_4 = [KPWD_M]_5 = \overline{[KPWD_M]_F}, \quad [KPWD_L]_0 = \overline{[KPWD_L]_3},$$
$$[KPWD_L]_A = [KPWD_L]_1; [KPWD_L]_4 = [KPWD_L]_5 = [KPWD_L]_F$$

Correlations obtained by considering $RN_2^{Tag}$ and $RN_4^{Tag}$, and XORing the relevant bits in $CCPwd_{L1}$ and $CCPwd_{L2}$:

$$[KPWD_M]_5 = \overline{[KPWD_M]_E}, \quad [KPWD_M]_2 = \overline{[KPWD_M]_D},$$
$$[KPWD_M]_A = \overline{[KPWD_M]_7}, \quad [KPWD_L]_5 = [KPWD_L]_E,$$
$$[KPWD_L]_2 = [KPWD_L]_D, \quad [KPWD_L]_A = [KPWD_L]_7$$

### Eavesdropped Session 2:

$RN_1^{Tag} = 0x41AB,\ RN_2^{Tag} = 0x9E41,\ RN_3^{Tag} = 0xC7FC,\ RN_4^{Tag} = 0x01F0$

Correlations obtained by considering $RN_1^{Tag}$ and $RN_3^{Tag}$, and XORing the relevant bits in $CCPwd_{M1}$ and $CCPwd_{M2}$:

$$[KPWD_M]_4 = \overline{[KPWD_M]_C} = \overline{[KPWD_M]_B}; [KPWD_M]_1 = [KPWD_M]_7,$$
$$[KPWD_M]_A = [KPWD_M]_F; [KPWD_L]_4 = [KPWD_L]_C = [KPWD_L]_B,$$
$$[KPWD_L]_1 = [KPWD_L]_7; [KPWD_L]_A = [KPWD_L]_F$$

Correlations obtained by considering $RN_2^{Tag}$ and $RN_4^{Tag}$, and XORing the relevant bits in $CCPwd_{L1}$ and $CCPwd_{L2}$:

$$[KPWD_M]_9 = [KPWD_M]_0 = [KPWD_M]_1, \quad [KPWD_M]_E = \overline{[KPWD_M]_1},$$
$$[KPWD_M]_4 = \overline{[KPWD_M]_F}, \quad [KPWD_L]_9 = [KPWD_L]_0 = \overline{[KPWD_L]_1},$$
$$[KPWD_L]_E = [KPWD_L]_1, \quad [KPWD_L]_4 = [KPWD_L]_F$$

Combining with the correlations obtained from the previous session:

$$\begin{aligned}[KPWD_M]_0 &= [KPWD_M]_1 = \overline{[KPWD_M]_3} = [KPWD_M]_4 = [KPWD_M]_5 \\ &= [KPWD_M]_7 = [KPWD_M]_9 = \overline{[KPWD_M]_A} = \overline{[KPWD_M]_B} \\ &= \overline{[KPWD_M]_C} = \overline{[KPWD_M]_E} = \overline{[KPWD_M]_F}, \\ [KPWD_M]_2 &= \overline{[KPWD_M]_D}, \quad [KPWD_L]_0 = \overline{[KPWD_L]_1} = \overline{[KPWD_L]_3} \\ &= \overline{[KPWD_L]_4} = \overline{[KPWD_L]_5} = \overline{[KPWD_L]_7} = [KPWD_L]_9 \\ &= \overline{[KPWD_L]_A} = \overline{[KPWD_L]_B} = \overline{[KPWD_L]_C} = \overline{[KPWD_L]_E} \\ &= \overline{[KPWD_L]_F}, \quad [KPWD_L]_2 = [KPWD_L]_D\end{aligned}$$

### Eavesdropped Session 3:

$RN_1^{Tag} = 0x08B6,\ RN_2^{Tag} = 0xEDF5,\ RN_3^{Tag} = 0x9A62,$
$RN_4^{Tag} = 0x38AF$

Correlations obtained by considering $RN_1^{Tag}$ and $RN_3^{Tag}$, and XORing the relevant bits in $CCPwd_{M1}$ and $CCPwd_{M2}$:

$$[KPWD_M]_0 = [KPWD_M]_9; [KPWD_M]_8 = \overline{[KPWD_M]_A},$$
$$[KPWD_M]_B = [KPWD_M]_6 = \overline{[KPWD_M]_2}; [KPWD_L]_0 = [KPWD_L]_9,$$
$$[KPWD_L]_8 = \overline{[KPWD_L]_A}; [KPWD_L]_B = [KPWD_L]_6 = [KPWD_L]_2$$

Combining with the correlations obtained from the previous session:

$$\begin{aligned}[KPWD_M]_0 &= [KPWD_M]_1 = [KPWD_M]_2 = \overline{[KPWD_M]_3} = [KPWD_M]_4 \\ &= [KPWD_M]_5 = \overline{[KPWD_M]_6} = [KPWD_M]_7 = [KPWD_M]_8 \\ &= [KPWD_M]_9 = \overline{[KPWD_M]_A} = \overline{[KPWD_M]_B} = \overline{[KPWD_M]_C} \\ &= \overline{[KPWD_M]_D} = \overline{[KPWD_M]_E} = \overline{[KPWD_M]_F}, \\ [KPWD_L]_0 &= \overline{[KPWD_L]_1} = \overline{[KPWD_L]_2} = \overline{[KPWD_L]_3} = \overline{[KPWD_L]_4} \\ &= \overline{[K_L]_5} = \overline{[KPWD_L]_6} = \overline{[KPWD_L]_7} = [KPWD_L]_8 \\ &= [KPWD_L]_9 = \overline{[KPWD_L]_A} = \overline{[KPWD_L]_B} = \overline{[KPWD_L]_C} \\ &= \overline{[KPWD_L]_D} = \overline{[KPWD_L]_E} = \overline{[KPWD_L]_F}\end{aligned}$$

At this point, without even considering $CCPwd_{L1}$ and $CCPwd_{L2}$, the adversary will have obtained the maximal (or full) set of correlations. The four possible values of $KPWD$ based on the correlations will be: $\{0x1237F3F, 0x123F80C0, 0xEDC07F3F, 0xEDC08DC0\}$.

## References


[1] S. Karthikeyan, M. Nesterenko, RFID security without extensive cryptography, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, 2005, pp. 63–67.

[2] D.N. Duc, J. Park, H. Lee, K. Kim, Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning, in: The 2006 Symposium on Cryptography and Information Security, 2006.

[3] H.Y. Chien, C.H. Chen, Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards, Computer Standards & Interfaces 29 (2007) 254–259.

[4] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, A. Ribagorda. Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard, in: Proceedings of Int'l Conference on RFID Security (RFIDSec)'07, Jul 2007.

[5] T.L. Lim, T. Li, Addressing the weakness in a lightweight RFID tag-reader mutual authentication scheme, in: Proceedings of the IEEE Int'l Global Telecommunications Conference (GLOBECOM) 2007, Nov 2007, pp. 59–63.

[7] D.M. Konidala, Z. Kim, K. Kim, A simple and cost-effective RFID tag-reader mutual authentication scheme, in: Proceedings of Int'l Conference on RFID Security (RFIDSec)'07, Jul 2007, pp. 141–152.

[8] D.M. Konidala, K. Kim, RFID Tag-reader mutual authentication scheme utilizing tag's access password, Auto-ID Labs White Paper WP-HARDWARE-033, Jan 2007.
[9] EPCglobal Inc., The EPCglobal architecture framework, EPCglobal standards, Jul 2005. Available from: <http://www.epcglobalinc.org/standards/>.
[10] EPCglobal Inc., EPC radio-frequency identity protocols Class-1 Generation-2 UHF RFID protocol for communications at 860MHz–960MHz Version 1.0.9, EPCglobal standards, Jan 2005. Available from: <http://www.epcglobalinc.org/standards/>.
[11] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, A. Ribagorda. RFID specification revised, in: The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, vol. 6, 2008, pp. 127–156.