Vulnerability Analysis of a Mutual Authentication Scheme under the EPC Class-1 Generation-2 Standard

Pedro Peris-Lopez[1], Tieyan Li[2], Tong-Lee Lim[2], Julio C. Hernandez-Castro[1], and Juan M. Estevez-Tapiador[1]

[1] Computer Science Department, Carlos III University of Madrid
{pperis, jcesar, jestevez}@inf.uc3m.es
[2] Institute for Infocomm Research, A*STAR Singapore
{litieyan,tllim}@i2r.a-star.edu.sg

Abstract. The security level of the EPC Class-1 Generation-2 RFID standard is very low, as shown in previous works such as [1–4]. In particular, the security of the access and kill passwords of an RFID tag is almost non-existent. A first initiative by Konidala and Kim [5] tried to solve these problems by proposing a tag-reader mutual authentication scheme (TRMA) to protect the tag access password. However, Lim and Li showed how a passive attacker can recover the access password of the tag [6]. Recently, Konidala and Kim proposed a new version of the TRMA scheme (TRMA$^+$) in which the tag access and kill passwords are used for authentication [7]. In this paper, we show that this new version still contains 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 higher than 2^{-5}. Finally, we show how an attacker can recover the entire kill password with probability 2^{-2} within 4 eavesdropped sessions in the case of a passive attack, or just 2 consecutive sessions under an active attack.

1 Introduction

Even a brief analysis of the EPC Class-1 Generation-2 standard reveals serious security problems. To overcome such weaknesses, some authors have proposed new schemes that, still under the standard framework, try to improve its security. In [1], an efficient tag identification and reader authentication protocol based on the use of XORs and matrix operations is proposed. 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 studied both protocols and showed that they contain certain security flaws [3]. As a result, they proposed a new scheme similar to that of Duc et al., though it later came under attack by Peris-Lopez et al. [4]. In 2007, Konidala, Kim and Kim proposed a new scheme intending to correct the security shortcomings of all previous proposals [7]. This protocol, which improves on an earlier version proposed in [5], is analyzed in this paper.

2 The Original TRMA Scheme and its Extension

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

2.1 Original TRMA Scheme

In [5], Konidala and Kim proposed an authentication scheme in an attempt to correct the security shortcomings discovered in the EPC-C1G2 standard. The authors claimed that the proposed scheme protects the tag 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 in [5].

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. $PAD_i = PadGen(RN_i^{Tag}, RN_i^{Reader})[APWD]$, where $PadGen(.)$ is a specially designed pad generation function. Next, two 16-bit random numbers (RN_3^{Rdr}, RN_4^{Rdr}), 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 [6], Lim and Li uncovered some 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 some correlation analysis on the captured information. In [7], Konidala and Kim proposed an improved version that uses the tag's access and kill passwords. The authors 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 in what follows. Let us represent the 32-bit $XPWD$ (where $XPWD \in \{APWD,\ KPWD\}$) in binary (or Base 2) as

$$\begin{aligned} XPWD &= XPWD_M \mid\mid XPWD_L \\ XPWD_M &= b_0 b_1 b_2 .....b_{13} b_{14} b_{15} \\ XPWD_L &= b_{16} b_{17} b_{18} .....b_{29} b_{30} b_{31} \end{aligned}$$

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

$$\begin{aligned} RN_i^{Tag} &= H_{i,0}^{Tag} H_{i,1}^{Tag} H_{i,2}^{Tag} H_{i,3}^{Tag} \\ RN_i^{Rdr} &= H_{i,0}^{Rdr} H_{i,1}^{Rdr} H_{i,2}^{Rdr} H_{i,3}^{Rdr} \end{aligned}$$

where each $H_{i,j}^{Tag}$ and $H_{i,j}^{Rdr}$ is a hexadecimal digit, i.e. $H_{i,j}^{Tag},\ H_{i,j}^{Rdr} \in \boldsymbol{H}_{16} = \{0\text{x}0, 0\text{x}1, 0\text{x}2, ..., 0\text{xD} = 13, 0\text{xE} = 14, 0\text{xF} = 15\}$.

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

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

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

As an example, let us consider $PadGen(\text{7E2B}_h, \text{2B5F}_h)[XPWD]$ with $XPWD_M = 1110\ 0101\ 0100\ 1000_2$ and $XPWD_L = 1110\ 1000\ 1100\ 1010_2$.

- $\text{7E2B}_h = 7^{th}\ 14^{th}\ 2^{nd}\ 11^{th}$ location of $XPWD_M = 1010_2$
- $\text{7E2B}_h = 7^{th}\ 14^{th}\ 2^{nd}\ 11^{th}$ location of $XPWD_L = 0110_2$
- $\text{2B5F}_h = 2^{nd}\ 11^{th}\ 5^{th}\ 15^{th}$ location of $XPWD_M = 1010_2$
- $\text{2B5F}_h = 2^{nd}\ 11^{th}\ 5^{th}\ 15^{th}$ location of $XPWD_L = 1000_2$

Combining the 4 results above, we have a 16-bit pad value $PadGen(\text{7E2B}_h, \text{2B5F}_h)[XPWD] = 1010\ 0110\ 1010\ 1000 = \text{A6A8}_h$

3 Attacks on TRMA$^+$

In this section we describe how an attacker can obtain an advantage towards recovering the 32-bit access and kill passwords for a tag under the TRMA$^+$ scheme.

3.1 Access Password Attack (LSB)

The attack is schematically outlined in the following figure. Details are provided below.

(1)	$Tag \rightarrow Reader$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2)		....
(3)	$Attacker \rightarrow Reader$:	$\{EPC, RN_1^{Tag'}, RN_2^{Tag'}\}$
(4)	$Reader \rightarrow Attacker$:	$\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$

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 obtained EPC, the adversary performs an active attack by masquerading as the target tag and participating in the TRMA$^+$ protocol with a genuine reader. The adversary sends the message $\{EPC, RN_1^{Tag'}, RN_2^{Tag'}\}$ 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 $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \tag{7}$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \tag{8}$$

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

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

Let $PadGen(RN_i^{Tag'}, RN_i^{Rdr})[APWD] = V_0V_1V_2V_3$ 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. $P_2, P_3 \in \{0000_b = 0_h,\ 1111_b = \mathrm{F}_h\}$. This leads to $P_2P_3 \in \{00_h, 0\mathrm{F}_h, \mathrm{F}0_h, \mathrm{FF}_h\}$. 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:

$$APWD_M[8...15] = \begin{cases} CCPwd_{M1}[8...15] \oplus \text{0x00} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0x0F} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xF0} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xFF} & \text{with } p = 2^{-2} \end{cases} \quad (10)$$

$$APWD_L[8...15] = \begin{cases} CCPwd_{L1}[8...15] \oplus \text{0x00} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0x0F} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0xF0} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0xFF} & \text{with } p = 2^{-2} \end{cases} \quad (11)$$

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

$$\begin{aligned} &APWD_M[8...15] \;||\; APWD_L[8...15] \\ &= \begin{cases} CCPwd_{M1}[8...15] \oplus \text{0x00} \;||\; CCPwd_{L1}[8...15] \oplus \text{0x00} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0x0F} \;||\; CCPwd_{L1}[8...15] \oplus \text{0x0F} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xF0} \;||\; CCPwd_{L1}[8...15] \oplus \text{0xF0} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xFF} \;||\; CCPwd_{L1}[8...15] \oplus \text{0xFF} & p = 2^{-2} \end{cases} \end{aligned} \quad (12)$$

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

3.2 Access Password Attack (MSB)

The attack (man-in-the-middle) is schematically outlined in the following figure. Details are provided below.

(1) $Tag \rightarrow Attacker$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2a) $Attacker \rightarrow Reader$:	$\{EPC, RN, RN\}$
(2b) $Reader \rightarrow Attacker$:	$\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$
(3a) $Attacker \rightarrow Reader$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(3b) $Reader \rightarrow Attacker$:	$\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr'}, RN_2^{Rdr'}, RN_3{}^{Rdr'}, RN_4{}^{Rdr'}\}$
(4) $Attacker \rightarrow Tag$:	$\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr'}, RN_2^{Rdr'}, RN, RN\}$
(5) $Tag \rightarrow Attacker$:	$\{RN_3^{Tag}, RN_4^{Tag}, CCPwd_{M2}, CCPwd_{L2}\}$

Scenario: An adversary intercepts and alters the content of the message sent by a genuine tag. The random numbers picked up by the adversary are then set to RN before forwarding 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: $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \tag{14}$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \tag{15}$$

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

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

In a different, parallel authentication session, the adversary forwards the initial message sent by the tag $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$ to the legitimate reader. The reader's response $\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr'}, RN_2^{Rdr'}, RN_3^{Rdr'}, RN_4^{Rdr'}\}$ 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: $\{CCPwd_{M2}, CCPwd_{L2}, RN_3^{Tag}, RN_4^{Tag}\}$, where

$$CCPwd_{M2} = APWD_M \oplus PAD_3 \tag{17}$$

$$CCPwd_{L2} = APWD_L \oplus PAD_4 \tag{18}$$

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

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

Under such an attack scenario, the adversary can derive the following:

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

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

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's described in *Section 3.1*). Then,

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

Assuming that the values of V_0 and V_1 are taken randomly from the set $\{0_h, \mathrm{F}_h\}$, then they would 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 would have $k_{V_0} = k_{V_1}$ with probability 0.5. Hence,

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

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

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

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

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

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

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 would $Prob(k_{S_2} = k_{S_3}) = 0.75$ 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

$$\begin{aligned}
k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3} \quad & p = 0.75 \\
k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3} \quad & p = 0.25
\end{aligned}$$

and

$$\begin{aligned}
k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16} \quad & p = 0.75 \\
k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} = k_{S_3+16} \quad & p = 0.25
\end{aligned}$$

However, instead of considering these two sets of relations separately, we shall combine them to give us 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 $S_2 \neq S_3$). Under 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

$$APWD_M[0...7] \,||\, APWD_L[0...7] = A \,\oplus\, mask \,||\, B \,\oplus\, mask \tag{25}$$

where

$$A = (CCPwd_{M1}[0..7] \,\wedge\, 0\text{xCC}) \,\vee\, (CCPwd_{M2}[0..7] \,\wedge\, 0\text{x33}) \tag{26}$$

$$B = (CCPwd_{L1}[0..7] \,\wedge\, 0\text{xCC}) \,\vee\, (CCPwd_{L2}[0..7] \,\wedge\, 0\text{x33}) \tag{27}$$

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 $mask \in \{0\text{x00}, 0\text{x0F}, 0\text{xF0}, 0\text{xFF}\}$. 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 those 16 bits of the access password would be

$$\begin{aligned} &Prob(\text{successful recovery of all bits in } APWD_M[0...7] \,||\, APWD_L[0...7]) \\ &= 0.625 \times 1/4 \\ &= 0.15625 \end{aligned} \tag{28}$$

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

In summary, with equations (25), (26) and (27) the probability of a successful attack for any selected mask would be given by

$$Prob(\text{successful recovery of all bits in } APWD_M[0...7] \,||\, APWD_L[0...7])$$

$$= \begin{cases} \frac{5}{2^5} = 0.15625 & \text{if } mask \in \{0\text{x00}, 0\text{x0F}, 0\text{xF0}, 0\text{xFF}\} \\ \frac{1}{2^5} = 0.03125 & \text{if } mask \in \{0\text{x05}, 0\text{x0A}, 0\text{x50}, 0\text{x55}, 0\text{x5A}, 0\text{x5F}, 0\text{xA0}, 0\text{xA5}, \\ & \quad 0\text{xAA}, 0\text{xAF}\ 0\text{xF5}, 0\text{xFA}\} \end{cases}$$

Hence, in order to maximize the probability of success of an attack, the adversary should select $mask$ from the set $\{0\text{x00}, 0\text{x0F}, 0\text{xF0}, 0\text{xFF}\}$. In any case, this attack results in 16 possible values for the most significant bits of $APWD_M$ and $APWD_L$. Together with the 4 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.

3.3 Kill Password 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 full 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:

$$PAD_1[8...15] = CCPwd_{M1}[8...15] \oplus APWD_M[8...15] \quad (29)$$
$$PAD_2[8...15] = CCPwd_{L1}[8...15] \oplus APWD_L[8...15] \quad (30)$$
$$PAD_3[8...15] = CCPwd_{M2}[8...15] \oplus APWD_M[8...15] \quad (31)$$
$$PAD_4[8...15] = CCPwd_{L2}[8...15] \oplus APWD_L[8...15] \quad (32)$$

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

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

Hence,

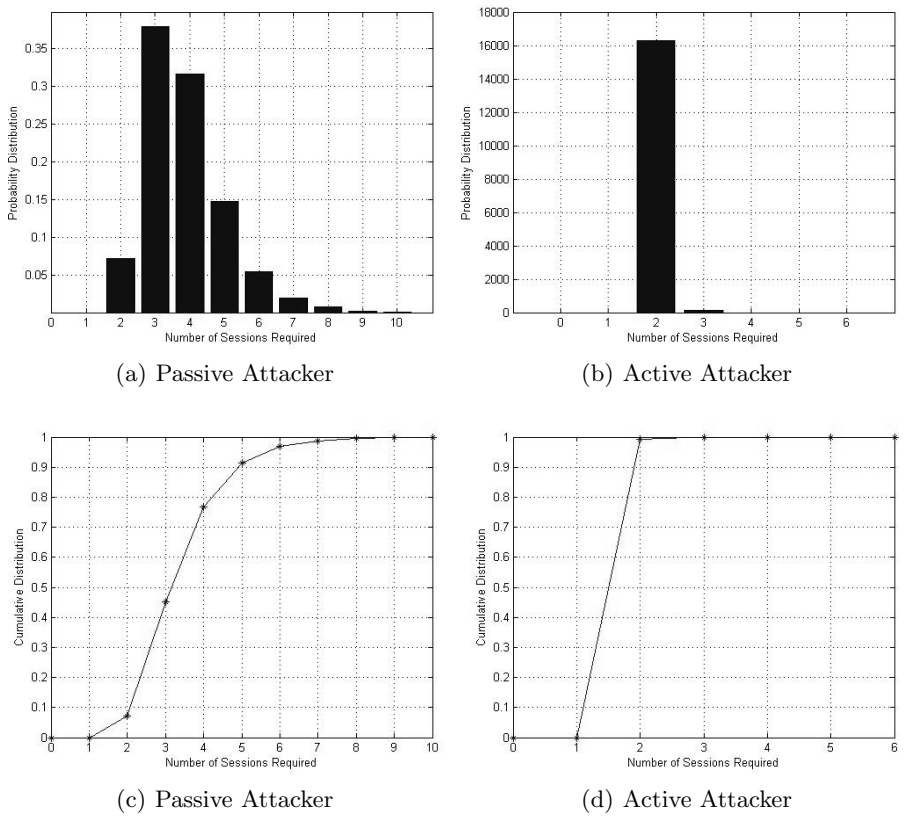
$$\begin{aligned} PAD_i[8...15] = k_{H_{i,0}^{Tag}}k_{H_{i,1}^{Tag}}k_{H_{i,2}^{Tag}}k_{H_{i,3}^{Tag}} \ || \\ k_{H_{i,0}^{Tag}+16}k_{H_{i,1}^{Tag}+16}k_{H_{i,2}^{Tag}+16}k_{H_{i,3}^{Tag}+16} \quad [Base\ 2] \end{aligned}$$

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

$$\begin{aligned} k_{H_{i,0}^{Tag}} &= PAD_i[8] & k_{H_{i,0}^{Tag}+16} &= PAD_i[12] \\ k_{H_{i,1}^{Tag}} &= PAD_i[9] & k_{H_{i,1}^{Tag}+16} &= PAD_i[13] \\ k_{H_{i,2}^{Tag}} &= PAD_i[10] & k_{H_{i,2}^{Tag}+16} &= PAD_i[14] \\ k_{H_{i,3}^{Tag}} &= PAD_i[11] & k_{H_{i,3}^{Tag}+16} &= PAD_i[15] \end{aligned}$$

where each bit $PAD_i[n]$ can be computed using one of equations (29), (30), (31) or (32). 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 would be able to obtain the bits in the kill password. For a complete 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 the full 32 bits of the kill password. Based



(a) Passive Attacker (b) Active Attacker

(c) Passive Attacker (d) Active Attacker

Fig. 1: Probability and cumulative distributions for the number of sessions required for a successful attack.

on our experiments, in which we simulated 20,000 executions of the attack, we find that the average number of sessions required to obtain the full kill password is 4. *Fig. 1(a)* and *1(c)* show the probability distribution and the cumulative distribution 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 full 32 bits of the kill password would be reduced. From our experimental results, we find 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. 1(b)* and *1(d)* show the results.

4 Conclusion

In this paper, we have shown the existence of some security weaknesses of a lightweight mutual authentication scheme presented at RFIDSec 2007. This protocol uses the access and kill password defined in the EPC 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 find that after some computations an attacker can acquire the access and kill passwords with high probability. The most and least significant eight bits of the access password can be obtained with a probability of 2^{-5} and 2^{-2}, respectively. Once the attacker knows the LSB of the access password, the 32-bits of the kill password can be derived with a probability of 2^{-2}. The efficiency of the attack on the kill password depends on whether the attack is passive or active. A passive attacker has to eavesdrop an average of 4 protocol rounds, and this number is reduced to 2 when the attacker can modify and manipulate the exchanged messages. In summary, we find that the security of EPC-C1G2 standard is too low. So far, most of the proposals that aim to increase its security while being compliant with the standard have failed. Incorporating greater security in future standards is indeed a much-needed challenge.

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*, pp. 63-67, 2005.
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", in *Computer Standards & Interfaces 29 (2007)*, pp. 254 - 259, 2007.
4. P. Peris-Lopez, J.C. Hernandez-Castro, J.M.E. Tapiador, and A. Ribagorda, "Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 standard", in *Conference on RFID Security, RFIDsec'07*, Malaga, Spain, July 2007.
5. D.M. Konidala and K. Kim, "RFID Tag-Reader Mutual Authentication Scheme Utilizing Tag's Access Password", *Auto-ID Labs White Paper WP-HARDWARE-033*, Jan 2007.
6. T.L. Lim, and 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*, pp. 59-63, Nov 2007.
7. D.M. Konidala, Z. Kim, and K. Kim, "A Simple and Cost-effective RFID Tag-Reader Mutual Authentication Scheme", in *Proceedings of Int'l Conference on RFID Security 2007 (RFIDSec 07)*, pp. 141-152, Jul. 11-13, 2007, Malaga, Spain.