This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

Analysis of RFID Mutual Authentication Protocols

Charng Rang Guo
Infocomm Infrastructure Programme Centre
Defence Science & Technology Agency
Singapore

***Abstract*—In order to secure RFID transactions, an authentication scheme has originally been included as part of the "EPC Gen2" industrial standard. This scheme, however, has been found to have some weaknesses. To overcome such weaknesses, some authors have proposed new schemes to improve its security. Two of the latest proposals are the Lim-Li protocol and the Konidala-Kim-Kim protocol presented in 2007. We propose an attack that can be applied to both these protocols. In both cases, the Search Space for the password(s) used have been drastically reduced, which effectively breaks the security of these protocols.**

Keywords-RFID, mutual authentication, protocol

I. INTRODUCTION

In order to secure RFID transactions, an authentication scheme has originally been included as part of the "EPC Gen2" industrial standard [1]. This scheme, however, has been found to have some weaknesses. To overcome such weaknesses, some authors have proposed new schemes to improve its security. These include, for example, the proposal in [2], which uses XOR and matrix operations, and the proposal in [3] which suggested Tag-to-backend database authentication. Both these protocols, however, are shown to have certain security flaws by Chien and Chen in [4]. Chien and Chen in turn proposed a new scheme to improve the security. On the other hand, their protocol was later found to be unsatisfactory in [5] as well.

In Jan 2007, Konidala and Kim proposed another scheme [6]. Later in that year, Konidala, Kim and Kim proposed a revised scheme [8] which improved the security of the original version. Unfortunately, both these protocols are found to be inadequate by Lim, Li et al. ([7] and [9]). In [7], besides presenting an analysis, Lim and Li also proposed additional enhancements to strengthen the protocol in [6].

This paper discusses the enhancement proposed by Lim and Li in [7] and the revised scheme proposed by Konidala, Kim and Kim in [8]. We propose an active attack that can be applied to both the Lim-Li protocol (in [7]) and the Konidala-Kim-Kim protocol (in [8]):

(1) We show that the Lim-Li protocol has weaknesses. To elaborate, the security of the protocol relies on a 32-bit Access Password. We show that, with 16 spoofed messages, an adversary is able to reduce the Search Space (i.e. the number of guesses of this password) from 2^{32} possibilities to 16 possibilities. This effectively breaks the security of the protocol. Moreover, our simulation on a refinement of the attack also shows that for 99.5% of the passwords, an adversary is able to determine the key used uniquely.

(2) For the Konidala-Kim-Kim protocol in [8], both the 32-bits Access Password and the 32-bits Kill Password are used. We show that, using a man-in-the-middle attack, with 18 spoofed messages, an adversary can reduce the possible guesses of the Access Password and Kill Password from a total of 2^{64} to 4. This is an improvement over the result in [9] in terms of the accuracy of the guesses.

This paper will be organized as follows. The notation used in this paper will be listed in the next section. The Lim-Li protocol in [7] and Konidala-Kim-Kim protocol in [8] will be presented in Section III. Section IV and V discuss the analysis of these two protocols respectively. The paper will be concluded in Section VI.

II. NOTATION

We follow some of the notation used in [8]. The notation *APwd* and *KPwd* denote the Tag's Access and Kill Password respectively. We also use $APwd_M$ and $APwd_L$ to denote the16 most significant and least significant bits of *APwd* respectively; and $CCPwd_M$ and $CCPwd_L$ to denote the Cover-Coded $APwd_M$ and $APwd_L$ respectively.

The symbol $\oplus$ denotes bit-wise XOR of two numbers, and the symbol $\|$ denotes concatenation of two strings.

For a 16-bit number X, we let $X[j]$ denote the jth bit of X ($X[0]$ being the most significant), and $X[j...k]$ denote

$$X[j] \,\|\, X[j+1] \,\|\, ... \,\|\, X[k\text{-}1] \,\|\, X[k].$$

Suppose we write

$$X = X_0\, X_1\, X_2\, X_3 = X_0\, X_1\, X_2\, X_3 \text{ [Base 16]}.$$

Then the hexadecimal digit X_j is the same as the number represented by $X[4j...4j+3]$, $j = 0,...,3$. Suppose we write

$$X = u_0\, u_1 ... u_{14}\, u_{15} = u_0\, u_1 ... u_{14}\, u_{15} \text{ [Base 2]}.$$

Then the bit u_j is the same as $X[j]$.

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

III. THE PROTOCOLS

Both the protocols in [7] and [8] concern a challenge-and-response authentication between a RFID Reader R and a Tag T. There is a 32-bits Access Password *APwd* and a 32-bits Kill Password *KPwd* that are shared between the Reader and the Tag. The protocol in [7] uses only the former password, whereas the protocol in [8] uses both parameters. The main idea for the protocols is as follows:

Tag → Reader	: EPC, RN_1^{Tag}, RN_2^{Tag}
Reader → Tag	: RN_1^{Rdr}, RN_2^{Rdr}, $CCPwd_{M1}$, $CCPwd_{L1}$, RN_3^{Rdr}, RN_4^{Rdr}
Tag	: Verify $CCPwd_{M1}$, and $CCPwd_{L1}$. Proceed only if successful
Tag → Reader	: RN_3^{Tag}, RN_4^{Tag}, $CCPwd_{M2}$, and $CCPwd_{L2}$
Reader	: Verify $CCPwd_{M2}$, and $CCPwd_{L2}$. Proceed only if successful.

Here EPC is a number in clear, RN_j^{Tag} and RN_j^{Rdr} are the nonce (sent by the Tag and the Reader respectively); and $CCPwd_{M1}$, $CCPwd_{L1}$, $CCPwd_{M2}$, and $CCPwd_{L2}$ are the responses to the challenges constructed as follows:

$$CCPwd_{M1} = APwd_M \oplus PAD_1,$$

$$CCPwd_{L1} = APwd_L \oplus PAD_2,$$

$$CCPwd_{M2} = APwd_M \oplus PAD_3,$$

$$CCPwd_{L2} = APwd_L \oplus PAD_4.$$

The values PAD_j's constitute the main difference between the Lim-Li protocol and the Konidala-Kim-Kim protocol. All these values use a complex function PG which takes in two 16-bits inputs and a password (either the Access Password or the Kill Password) to produce a 16-bit output (in other words, for X and Y and a password $XPwd$, $PG(X, Y)[XPwd]$ is a 16-bit value). We shall describe this function in details later. For the Konidala-Kim-Kim protocol, for $i = 1,\ldots 4$,

$$PAD_i = PG\,(PG\,(RN_i^{Tag}, RN_i^{Rdr})\,[APwd], RN_i^{Tag})\,[KPwd].$$

For the Lim-Li protocol, a few variants are presented in [7]. To fix ideas, we present one of them here. (The other variants can be similarly analyzed.) In this variant, for $j = 1, \ldots, 4$,

$$PAD_j = PG\,(U_j, V_j)\,[APwd],$$

$$U_j = RN_j^{Tag} \oplus APwd_M,\ V_j = RN_j^{Rdr} \oplus APwd_M.$$

The function PG is a function that mixes the inputs X and Y. Following the description in [9], let us represent the 32-bit $XPwd$ (where $XPwd \in \{APwd, KPwd\}$) in binary

$$XPwd = XPwd_M \,||\, XPwd_L,$$

$$XPwd_M = b_0 b_1 \ldots b_{14} b_{15},\ XPwd_L = b_{16} b_{17} \ldots b_{30} b_{31}.$$

Also, let us represent the 16 bit random numbers X and Y in hexadecimal (or Base 16) representation as

$$X = X_0\, X_1\, X_2\, X_3 \text{ [Base 16]},$$

$$Y = Y_0\, Y_1\, Y_2\, Y_3 \text{ [Base 16]}.$$

The function $PG(X, Y)[XPwd]$ is defined as follows:

$$PG(X,Y)[XPwd] = b_{X0} b_{X1} b_{X2} b_{X3} || b_{X0+16} b_{X1+16} b_{X2+16}\, b_{X3+16} ||$$
$$b_{Y0} b_{Y1} b_{Y2} b_{Y3} ||\, b_{Y0+16} b_{Y1+16} b_{Y2+16} b_{Y3+16} \text{ [Base 2]}$$

IV. ANALYSIS OF THE LIM-LI PROTOCOL

A. To reduce the Search Space

As an overview to the attack, we first note that an adversary may spoof as the Tag and send challenges to the legitimate Reader. From the first two steps of the protocol, we see that the Reader will promptly reply to the challenge without doing any check:

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

The adversary may thus collect data relating to the passwords ($CCPwd_{M1}$ and $CCPwd_{L1}$) from the reply. He may discontinue the session and repeat the process to collect more information. When sufficient information has been collected, he can then perform analysis and computation to reduce the Search Space.

As our attack only involves RN_1^{Tag}, RN_1^{Rdr}, and $CCPwd_{M1}$, for ease of presentation, we will look at the following simpler protocol for the rest of the discussion in this section.

T → R: a

R → T: b, R

Here the response R is equal to $M \oplus PG\,(X, Y)[A]$, where the parameter $A = M \,||\, L$ is the Access Password, X is equal to the XOR $a \oplus M$, and Y is equal to $b \oplus M$.

In our attack, the adversary initiates 16 spoofed sessions with the challenge $a = a^{(j)}$, where

$$a^{(j)} = j\,j\,j\,j \text{ [Base 16]}, \qquad j = 0, 1, 2, \ldots, 15.$$

We let $b^{(j)}$ and $R^{(j)}$, $j = 0,\ldots 15$ denote the replies from the Reader. We will show that the adversary is able to reduce the guesses for the password using these data.

In order to do this, we represent M as $M_0 M_1 M_2 M_3$, M_j's are hexadecimal digits. The adversary first makes a guess of M_0. Let $m_0 m_1 m_2 m_3$ [Base 2] denote the guess he has made. We further let $X^{(j)}$ denote $a^{(j)} \oplus M$ for $j = 0, 1, \ldots, 15$. In this case, $X^{(j)}{}_0$ is equal to $j \oplus m_0 m_1 m_2 m_3$. When j runs through 0, 1, 2, ... to 15, we see that $X^{(j)}{}_0$ will take the 16 values 0, 1, 2,..., 15 as well but in a "randomized" order. The adversary takes notes of these 16 numbers $X^{(j)}{}_0$.

From the protocol, the adversary also knows that the most significant bit $R^{(j)}[0]$ of $R^{(j)}$ is the XOR of the most significant bit $M[0]$ of M and the most significant bit of $PG\,(X, Y)[A]$. From the definition of PG, we can see that this latest bit is equal to $M[X^{(j)}{}_0]$. In other words,

$$R^{(j)}[0] = M[0] \oplus \mathrm{M}[X^{(j)}{}_0],$$

This implies that

$$\mathrm{M}[X^{(j)}{}_0] = M[0] \oplus R^{(j)}[0].$$

Since he has captured the value $R^{(j)}[0]$, from his guess that $M[0]$ is equal to m_0, he is able to compute $\mathrm{M}[X^{(j)}{}_0]$, $j = 0, \ldots, 15$. From these computed values and the 16 numbers $X^{(j)}{}_0$ he has

computed earlier, he will be able to obtain $M[k]$ for $k = 0, ..., 15$. In other words, he will be able to obtain the value of M based on his guess on M_0.

He may repeat the above process with another choice of M_0. This will lead to another value of M. As there are 16 possibilities for M_0, he will eventually obtain 16 possible values for M. Note that one of these will be the correct M in use.

For each of the guesses for the value M, the adversary may proceed to determine the value L. To do this, we note that the fifth bit $R^{(j)}[4]$ of $R^{(j)}$ is the XOR of the fifth bit $M[4]$ of M and the fifth bit of $PG(X, Y)[A]$. From the definition of PG, we can see that this latest bit is equal to $L[X^{(j)}{}_0]$. Thus we have

$$R^{(j)}[4] = M[4] \oplus L[X^{(j)}{}_0].$$

This gives

$$L[X^{(j)}{}_0]. = M[4] \oplus R^{(j)}[4].$$

As the adversary has captured the value $R^{(j)}, j = 0, 1, ..., 15$, he will have the value $R^{(j)}[4], j = 0, 1, ..., 15$. This allows him to compute $M[4] \oplus R^{(j)}[4]$. These 16 values are equal to $L[X^{(j)}{}_0]$ for $j = 0, ..., 15$. We have noted above that $X^{(j)}{}_0 = j \oplus M_0$ and will take all 16 values 0, 1,.., 15 (in a possibly "randomized" order). This means that $L[X^{(j)}{}_0]$ will run through $L[j], j = 0,.., 15$. The adversary is thus able to obtain L. Again, we note that as long as the value M is correct, the correct value of L will be obtained as well.

Thus we see that, as there are 16 possible values of M_0, from the data captured, the adversary is able to reduce the Search Space for the password (M, L) from 2^{32} to 16.

B. *Identification of correct key*

We next discuss a refinement of the attack that may help to reduce the Search Space further. We recall that

$$X^{(j)} = a^{(j)} \oplus M \text{ for } j = 0, 1, \ldots, 15.$$

Hence, for $k = 1, 2, 3$,

$$X^{(j)}{}_k = (a^{(j)} \oplus M)_k = j \oplus M_k. \quad (1)$$

From (1), we note that, with the value of M calculated, the adversary is able to compute $X^{(j)}{}_k, j = 0,.., 15, k = 1, 2, 3$ as well. This will further allow him to compute

$$M[k] \oplus M[X^{(j)}{}_k], \ j = 0, 1,..., 15, k = 1, 2, 3.$$

From the formula

$$R^{(j)}[k] = M[k] \oplus M[X^{(j)}{}_k], k = 1, 2 ,3,$$

we see that the last set of values he has computed should be equal to the values $R^{(j)}[k]$ he has collected. Thus for a guess of the value M_0, an adversary is able to check for 52 equalities

(1) for $j = 0, 1, 2, 3$, whether the guess m_j is equal to the value $M[j]$ computed,

(2) for $j = 0, 1,...,15, k = 1, 2, 3$, whether the value $R^{(j)}[k]$ collected is equal to the value $M[k] \oplus M[X^{(j)}{}_k]$ computed.

Suppose the adversary has made the correct guess. In that case all the 52 equalities will be consistent. On the other hand, if he has made a wrong guess, intuitively the randomized $X^{(j)}{}_0$ computed and the subsequently computation may lead to inconsistency in some of the equalities. This approach allows an adversary to reduce the number of guesses for M further with certain degree of success. Our simulation for this refinement confirms that this is indeed the case. For 99.5% of the keys tried, our simulation shows that the key can be identified uniquely. Further research will need to be done to determine the key in all cases.

V. Analysis of the Konidala-Kim-Kim Protocol

We next look at the Kondidala-Kim-Kim Protocol. The attack of this protocol is similar to that presented in Section IV. On the other hand, the attack will be more complex. There are altogether four steps. We will try to obtain *KPwd* first, followed by obtaining various bits in *APwd* in stages.

A. *Step 1 Obtaining KPwd*

In this step of the attack, the adversary again spoofs as the Tag and initiates 16 spoofed sessions with the challenge $RN^{(j)}$, where

$$RN^{(j)} = j\,j\,j\,j \text{ [Base 16]}, j = 0, 1, 2,..., 15.$$

He will collect the response from the Reader to carry out analysis. This is shown in the protocol exchange below:

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

…

Attacker → Reader: $\{EPC, RN^{(j)}, RN^{(j)}\}$, where $RN^{(j)} = jjjj$,

Reader → Attacker: $\{CCPwd_{M1}^{(j)}, CCPwd_{L1}^{(j)}, RN_1^{Rdr(j)}, RN_2^{Rdr(j)}, RN_3^{Rdr(j)}, RN_4^{Rdr(j)}\}$

Thus by sending 16 spoofed messages, the adversary would have obtained the 16 sets of values $\{CCPwd_{M1}^{(j)}, CCPwd_{L1}^{(j)}\}$. We note that

$$CCPwd_{M1}^{(j)} = APwd_M \oplus PAD_1^{(j)}.$$

This implies that

$$PAD_1^{(j)} = APwd_M \oplus CCPwd_{M1}^{(j)},$$

which in turn implies that

$$PAD_1^{(j)}[8] = APwd_M[8] \oplus CCPwd_{M1}^{(j)}[8],$$

$$PAD_1^{(j)}[12] = APwd_M[12] \oplus CCPwd_{M1}^{(j)}[12].$$

The choice of the number 8 and 12 will be clear from the discussion below. Suppose the adversary makes a guess for $APwd_M[8]$ and $APwd_M[12]$. He would have the values for the expressions on the right-hand-side of the equations above. We shall show that these will allow him to obtain a possible choice of Kill Password *KPwd*. To see this, we represent *KPwd* as $k_0k_1...k_{30}k_{31}$ [Base 2]. We note that, for each j

$$PAD_1^{(j)} = PG(PG(RN^{(j)}, RN_1^{Rdr(j)})[APwd], RN^{(j)})[KPwd]$$

$$= PG(V_{0,j}V_{1,j}V_{2,j}V_{3,j}, jjjj)[KPwd]$$

for some $V_{0,j}, V_{1,j}, V_{2,j}, V_{3,j}$. Thus we see that $PAD_1^{(j)}$ is equal to

$$k_{V0,j}k_{V1,j}k_{V2,j}k_{V3,j} \,||\, k_{V0,j+16}k_{V1,j+16}k_{V2,j+16}k_{V3,j+16}|| \; k_jk_jk_jk_j \,||\, k_{j+16}k_{j+16}k_{j+16}k_{j+16}.$$

Hence $PAD_I^{(j)}$ [8] is equal to k_j and $PAD_I^{(j)}$ [12] is equal to k_{j+16}. As j runs through 0 to 15, $PAD_I^{(j)}$ [8] will run through k_0, .., k_{15} and $PAD_I^{(j)}$ [12] will run through k_{16}, .., k_{31}. Thus we see that the two equations

$$PAD_I^{(j)}\,[8] = APwd_M\,[8] \oplus CCPwd_{MI}^{(j)}\,[8],$$

$$PAD_I^{(j)}\,[12] = APwd_M\,[12] \oplus CCPwd_{MI}^{(j)}\,[12],$$

indeed give the value of *KPwd* for each guess of $APwd_M$ [8] and $APwd_M$ [12]. As there are 4 possibilities for the two bits $APwd_M$[8] and $APwd_M$[12], his computation will reduce the number of guesses for *KPwd* to 4.

With the value of *KPwd* we may next obtain $APwd_M$[9...11], $APwd_M$[13...15] and $APwd_L$[8...15]. For ease of presentation, we shall discuss the method to obtain the 16 bits $APwd_M$[8...15] and $APwd_L$[8...15]. The two bits $APwd_M$[8] and $APwd_M$[12] obtained will have the same values as the guesses the adversary has made.

B. Step 2 Obtaining $APwd_M$[8...15] and $APwd_L$[8...15]

In order to obtain these 16 bits, the adversary looks at the data he has collected from one of the spoofed session with the spoofed challenge $RN = RRRR$. (We will choose $R = 8$ later.) We recall that

$$CCPwd_{MI} = APwd_M \oplus PAD_1, \text{ and}$$

$$CCPwd_{LI} = APwd_L \oplus PAD_2.$$

This implies that

$$APwd_M = CCPwd_{MI} \oplus PAD_1, \text{ and}$$

$$APwd_L = CCPwd_{LI} \oplus PAD_2,$$

which in turn implies that

$$APwd_M\,[8...15] = CCPwd_{MI}\,[8...15] \oplus PAD_I[8...15], \text{ and}$$

$$APwd_L\,[8...15] = CCPwd_{LI}\,[8...15] \oplus PAD_2\,[8...15].$$

These two equations show that the adversary will be able to obtain the 16 bits $APwd_M$ [8...15] and $APwd_L$ [8...15] if he is able to determine the values PAD_I[8...15] and PAD_2[8...15]. This is indeed possible by studying the PAD_j's more closely. To see this, we again note that for $i = 1$ or 2, PAD_i is equal to

$$PG\,(PG(RRRR, RN_i^{Rdr})\,[APwd], RRRR)\,[KPwd]$$

$$= PG\,(V_{0,I}V_{1,I}V_{2,I}V_{3,I}, RRRR)\,[KPwd] \qquad (2)$$

for some $V_{0,i}$, $V_{1,i}$, $V_{2,i}$, $V_{3,i}$. As the values of V's are not important at this moment, for simplicity of presentation, we shall write them as V_0, V_1, V_2, V_3. The expression on the right-hand-side of (2) is thus equals to

$$k_{V0}k_{V1}k_{V2}k_{V3} \,||\, k_{V0+16}k_{V1+16}k_{V2+16}k_{V3+16}|| \; k_Rk_Rk_Rk_R \,||\, k_{R+16}k_{R+16}k_{R+16}k_{R+16}.$$

Hence we see that PAD_I [8…15] is equal to

$$k_Rk_Rk_Rk_R \,||\, k_{R+16}k_{R+16}k_{R+16}k_{R+16}.$$

Thus we see that, as the adversary knows the value of *KPwd* and *R*, he is above to obtain the value of PAD_I [8…15]. This implies he is indeed able to obtain the values for $APwd_M$[8…15] and $APwd_L$[8…15] as well.

C. Step 3 Obtaining $APwd_M$[0…1], $APwd_M$[4…5], $APwd_L$[0…1] and $APwd_L$[4…5]

This step of the attack is similar to Step 2. Following the analysis above, we see that, for $x = 0$ and $y = 1$, or $x = 4$ and $y = 5$,

$$APwd_M\,[x…y] = CCPwd_{M1}\,[x…y] \oplus PAD_I[x…y]$$

$$APwd_L\,[x…y] = CCPwd_{L1}\,[x…y] \oplus PAD_2\,[x…y].$$

From the discussion involving PAD_j above, we also see that

$$PAD_I\,[0…1] = k_{V0,\,I}\,k_{V1,\,I}$$

$$PAD_I\,[4…5] = k_{V0,\,I+16}\,k_{V1,I+16},$$

for some $V_{0,I}$, $V_{1,i}$. As the adversary already obtained the Kill Password, these equations suggest that, if he is able to obtain the values $V_{0,I}$, $V_{1,I}$, he would be able to obtain the 8 bits of Access Password.

To see how this can be done, we recall that

$$V_{0,I}V_{1,I}V_{2,I}V_{3,I} = PG(RRRR, RN_I^{Rdr})\,[APwd].$$

This implies that

$$V_{0,I}V_{1,I}V_{2,I}V_{3,I} = PG(RRRR, U_{0,I}U_{1,I}U_{2,I}U_{3,I})\,[APwd] \qquad (3)$$

for some $U_{0,i}$, $U_{1,i}$, $U_{2,i}$, $U_{3,i}$. Representing *APwd* as $a_0a_1...a_{30}a_{31}$ [Base 2], we note that the last expression on the right-hand-side of (3) is in turns equal to

$$a_R\,a_R\,a_R\,a_R \,||\, a_{R+16}\,a_{R+16}\,a_{R+16}\,a_{R+16}||$$

$$a_{U0,i}\,a_{U1,i}\,a_{U2,i}\,a_{U3,i} \,||\, a_{U0,i+16}\,a_{U1,i+16}\,a_{U2,i+16}\,a_{U3,i+16}.$$

In other words, we note that, for both $i = 1$ or 2, $V_{0,i}$ is equal to $a_Ra_Ra_Ra_R$ and $V_{1,i}$ is equal to $a_{R+16}a_{R+16}a_{R+16}a_{R+16}$. Suppose we let $R = 8$. In this case $V_{0,i}$ is equal to $a_8a_8a_8a_8$ and $V_{1,i}$ is equal to $a_{24}\,a_{24}\,a_{24}\,a_{24}$. As the adversary has obtained the value of $APwd_M$[8] and $APwd_L$[8] (which is equal to a_8 and a_{24} respectively), he will thus know the values for $V_{0,i}$ and $V_{1,i}$. From our discussion above, this gives $APwd_M$[0...1], $APwd_M$[4...5], $APwd_L$[0...1] and $APwd_L$[4...5].

D. Step 4 Obtaining $APwd_M$[2...3], $APwd_M$[6...7], $APwd_L$[2...3] and $APwd_L$[6...7].

The last step of his attack to obtain the remaining 8 bits of passwords is more complex. The information obtained from the previous 16 exchanges is no longer sufficient. The adversary will need to launch a man-in-the-middle attack similar to that in [9] as follow (the nonce *RN* in the protocol will take the value 8 eventually):

Tag → Attacker: $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
Attacker → Reader: $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
Reader → Attacker: $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$
Attacker → Tag: $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN,RN\}$
Tag → Attacker: $\{CCPwd_{M2}, CCPwd_{L2}, RN_3^{Tag}, RN_4^{Tag}\}$

In other words, the adversary will first spoof as the Tag to obtain the response from the Reader. He will modify the message from the Reader (by replacing the challenges sent by the Reader) before sending it to the Tag. As the value $CCPwd_{M1}$ and $CCPwd_{L1}$ are computed by the Reader, the verification by the Tag will be successful. The Tag will revert with the response $CCPwd_{M2}$, $CCPwd_{L2}$. These contain the information the adversary needs to obtain the remaining 8 bits of $APwd$.

The analysis follows the approach in the last few sections. To see this, we note that

$$APwd_M = CCPwd_{M2} \oplus PAD_3$$

$$APwd_L = CCPwd_{L2} \oplus PAD_4.$$

In particular, for $x = 2$ and $y = 3$, or $x = 6$ and $y = 7$,

$$APwd_M[x...y] = CCPwd_{M2}[x...y] \oplus PAD_3[x...y]$$

$$APwd_L[x...y] = CCPwd_{L2}[x...y] \oplus PAD_4[x...y].$$

We also note that, for $i = 3$ or 4, PAD_i is equal to

$$PG\,(PG(RN_i^{Tag}, RN\ RN\ RN\ RN)\ [APwd], RN_i^{Tag})\ [KPwd]$$

We observe that $PG(RN_i^{Tag}, RN\ RN\ RN\ RN)\ [APwd]$ is equal to $S_0S_1S_2S_3$ for some S_j's, where S_2 is equal to $a_Ra_Ra_Ra_R$ and S_3 is equal to $a_{R+16}a_{R+16}a_{R+16}a_{R+16}$. (Here, again, we represent $APwd$ as $a_0...a_{31}$ [Base 2].) Hence we see that for each i, PAD_i is equal to

$$PG\,(S_0S_1S_2S_3, W_0W_1W_2W_3)\ [KPwd]$$

for some W_j's. The last expression is in turn equal to

$$k_{S0}\,k_{S1}\,k_{S2}\,k_{S3} \,||\, k_{S0+16}\,k_{S1+16}\,k_{S2+16}\,k_{S3+16} \,||$$

$$k_{W0}\,k_{W1}\,k_{W2}\,k_{W3} \,||\, k_{W0+16}\,k_{W1+16}\,k_{W2+16}\,k_{W3+16}.$$

In other words, we note that PAD_i [2...3] is equal to $k_{S2}k_{S3}$ and PAD_i [6...7] is equal to $k_{S2+16}\,k_{S3+16}$. Suppose we again let RN takes the value of 8. In this case, we see that S_2 is equal to $a_8a_8a_8a_8$ and S_3 is equal to $a_{24}a_{24}a_{24}a_{24}$. Following the argument above, as the adversary has obtained the value of $APwd_M[8]$ and $APwd_L[8]$, he will thus know the values for S_2 and S_3. From his knowledge of the Kill Password $KPwd$, he will obtain the values k_{S2}, k_{S3}, k_{S2+16}, k_{S3+16}, and consequently the values of PAD_i [2...3] and PAD_i [6...7]. This gives $APwd_M[2...3]$, $APwd_M[6...7]$, $APwd_L[2...3]$ and $APwd_L[6...7]$.

VI. Conclusion

In this paper, we show that two recently proposed protocols, the Lim-Li protocol in [7] and the Konidala-Kim-Kim protocol in [8], have security weaknesses. We propose an attack that can be applied to both schemes and drastically reduced the Search Space for the password(s) used. The Search Space of the Lim-Li protocol is reduced from 2^{32} to 16, whereas that for the Konidala-Kim-Kim protocol is reduced from 2^{64} to 4. This shows that the problem of designing an effective mutually authentication protocol for RFID transactions is indeed a difficult one. More research will be needed before we obtain a satisfactory scheme.

References

[1] EPCglobal Inc., "EPC radio-frequency identity protocols Class 1 Generation-2 UHF RFID protocol for communications at 860MHz-960MHz version 1.0.9", EPCglobal Stardards, Jan 2005.

[2] 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.

[3] 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.

[4] H.Y. Chien, C. H. Chen, "Mutual authentication protocol for RFID confirming to EPC Class 1 Generation 2 standards", in Computer Standards & Interfaces 29 (2007), pp. 254-259, 2007.

[5] P. Peris-Lopez, J.C. Hernadez-Castro, J.M.E. Tapiador, and A. Ribagorda, "Cryptanalysis of a novel authentication protocol conforming to EPC-CIG2 standard", in Conference on RFID Security, RFIDsec'07, Malaga, Spain, July 2007.

[6] D. M. Konidala and K. Kim, "RFID Tag-Reader mutual authentication scheme utilizing tag's Access Password", Auto-ID Labs White Paper WP-HARDWARE-003, Jan 2007.

[7] T L Lim and Tieyan Li, "Addressing the weakness in a lightweight RFID Tag-Reader mutual authentication scheme", Globecom07, Washington D.C., USA, Nov 2007.

[8] D.M. Konidala, Z. Kim and K. Kim, "A simple and cost-effective RFID Tag-Reader mutual authentication scheme", in Proceeding of Int'l Conference on RFID Security 2007 (RFIDSec 07), pp.141-152, Jul. 11-13, 2007, Malaga, Spain.

[9] P Peris-Lopez, T. Li, T-L Lim, J C. Hernandez-Castro, and J. M. Estevez-Tapiador, "Vulnerability analysis of a mutual authentication scheme under the EPC Class-1 Generation-2 standard", in Proceedings of 4th Workshop on RFID Security, 9-11 July 2008, Budapest, Hungary, pp. 52-63.