Passive Attacks on a Class of Authentication Protocols for RFID

Basel Alomair, Loukas Lazos, and Radha Poovendran

Network Security Lab.
Electrical Engineering Department
University of Washington
{alomair,llazos,rp3}@u.washington.edu

Abstract. Mutual authentication mechanisms can be used in RFID systems to preserve the confidentiality of the RFID tags. Hiding the unique IDs of the tags is critical to prevent unauthorized tag tracking. In this paper, we analyze two mutual authentication protocols called M^{2}AP and EMAP, recently proposed by Peris-Lopez et. al. We show that a *passive* adversary eavesdropping on the open wireless medium, can extract the unique ID of the RFID tag by collecting an expected $O(\log_2 L)$ challenge-response exchange messages between the tag and the reader, where L is the length of the tag's unique ID. To date, previously known attacks on M^{2}AP and EMAP require the active probing of each tag. Furthermore, attacks on M^{2}AP require $O(L)$ active queries to be sent to the tag by a rogue reader, as opposed to $O(\log_2 L)$.

Keywords: RFID, authentication, privacy, passive attack.

1 Introduction

Radio Frequency Identification (RFID) systems enable the unique identification of an item with an embedded RFID tag. Making use of radio frequency based technology, RFID tags can be scanned in a non-line-of-sight manner and can be batch processed [19]. Hence, RFID systems facilitate a variety of applications such as, supply chain management, inventory tracking, building access control, and smart home appliances.

RFID systems consist of three main components: RFID tags, RFID readers, and a database. To obtain the ID from an RFID tag, the reader requests for the tag's ID. The tag responds with a quantity that can be uniquely associated with its ID. The reader looks up the tag's ID in the database to obtain related information such as a detailed description of the product carrying the RFID tag. Unlike bar codes, RFID tags are associated with a unique identifier that can be linked to the individual product, not only to the product type. Since every tag carries a unique ID, tracking individual tags is feasible via a relatively low-cost RFID reader, thus compromising the privacy of the tag and eventually of the owner of the product [16]. As an example, scanning parked cars with an RFID reader can reveal which one has more valuables inside. Furthermore, an individual can be tracked simply by tracking the ID of any RFID tag he/she carries.

K.-H. Nam and G. Rhee (Eds.): ICISC 2007, LNCS 4817, pp. 102–115, 2007.

To prevent tracking of the tag by its unique ID, the tag must respond only to queries originated by authorized parties (readers). Furthermore, readers must have a mechanism to verify that any response to their queries comes only from the valid tags. These properties can be guaranteed if the reader and the tag can mutually authenticate one another.

While the problem of mutual authentication has well known solutions for computationally capable devices [18], it becomes particularly challenging in RFID systems due to the stringent hardware constraints of RFID tags. At present, a typical low-cost RFID tag has few thousands gates, in which only few hundreds of them can be dedicated to security [12]. Public Key Encryption (PKI) is beyond the computational power of the RFID tags due to the required exponentiations [12]. Even the symmetric encryption algorithms, like AES, typically require, on the order of, 20,000-30,000 gates [12][1], while cryptographic hash functions, such as SHA-1, are also too costly to be used in low-cost RFID tags [12]. In [1,2], Peris-Lopez et. al. have recently proposed two extremely lightweight protocols, called M^{2}AP and EMAP, where tags were assumed to have minimal computational power able to perform only bitwise XOR ($\oplus$), AND ($\wedge$), OR ($\vee$), and modulo addition operations. The basic idea behind M^{2}AP and EMAP is to use a temporary *index-pseudonym* (IDP) to hide the tag ID when communicating with a reader. The tag responds to the reader queries with an IDP, that can be linked to the tag's unique ID only by authorized readers.

Our contributions. In this paper, we analyze two lightweight mutual authentication protocols [1,2], called M^{2}AP and EMAP. We show how an adversary eavesdropping on the wireless channel can breach the confidentiality of the communication by extracting the tag's unique ID. Our attack model does not require the ability to modify the contents of transmitted messages, nor does it require the ability to actively probe tags; simple bitwise operations are sufficient to extract the unique ID of the tag. We provide probabilistic analysis of our attacks on both protocols and show that the problem of extracting the tag's ID can be mapped to a set cover problem. Our mapping shows that the number of protocol runs needed to extract the tag's unique ID is *logarithmic* in the length of the ID. Our attacks are *passive* and require eavesdropping of only $O(\log_2 L)$ protocol runs, as opposed to the *active* attacks presented in [3,4].

The rest of the paper is organized as follows. In Section 2, we state our assumptions. In Section 3, we describe the M^{2}AP and EMAP mutual authentication protocols. In Section 4, we describe attacks against the M^{2}AP and EMAP, and provide probabilistic analysis of our passive attacks against them. In Section 5, we present related work. We present our conclusions Section 6.

2 Adversarial Model

We assume a passive adversary able to eavesdrop on messages exchanged between legitimate RFID tag-reader pairs. We also assume that the adversary can store

[1] In [20], however, Feldhofer et al. described an AES implementations for RFID which requires about 3600 gates.

the messages it observes. Although a passive adversary is close to the weakest adversary one can have, our adversary, however, is a rather weak adversary as it only requires the ability to perform simple bit-wise operations and modulo additions. We do not consider an active adversary able to probe tags as in [3,4].

3 Description of the M^{2}AP and EMAP Protocols

3.1 The M^{2}AP Mutual Authentication Protocol

In the M^{2}AP protocol [1], each tag stores three quantities: the tag's secret unique ID, an IDP, and a secret key K=$K_1 \parallel K_2 \parallel K_3 \parallel K_4$, where $\parallel$ denotes the concatenation operation. For each tag, the IDP and secret key K are stored in the database. The tag's unique ID is static while the IDP and the key K are updated after every successful mutual authentication run. As a mutual authentication run, or protocol run, we define the execution of the following steps that lead to the mutual authentication of the reader-tag pair and the update of the IDP and K.

STEP 1: *Tag interrogation*–Initially, the reader sends a '*hello*' message to the tag which responds with its current IDP. Using the IDP, the reader retrieves the key K from the database.

STEP 2: *Reader authentication*–After receiving the IDP, and retrieving K, the reader generates two fresh random numbers (nonces), n_1 and n_2, and forwards the following three messages, A, B, and C in the clear, to the tag:

$$A = IDP \oplus K_1 \oplus n_1,\ B = (IDP \wedge K_2) \vee n_1,\ C = IDP + K_3 + n_2. \quad (1)$$

Upon receiving A, B, and C, the tag extracts the nonce n_1 from A as $n_1 = A \oplus IDP \oplus K_1$, and authenticates the reader by checking that $B = (IDP \wedge K_2) \vee n_1$. If authentication of B fails, the tag does not respond to the reader.

STEP 3: *Tag authentication*–After the reader has been authenticated, the tag extracts the nonce n_2 from message C as $n_2 = C - IDP - K_3$, and generates two messages, D and E, as follows:

$$D = (IDP \vee K_4) \wedge n_2, \quad E = (IDP + ID) \oplus n_1. \quad (2)$$

The reader authenticates the tag, by checking that $D = (IDP \vee K_4) \wedge n_2$.

STEP 4: *ID extraction*–The reader extracts the tag's unique ID from the message E as $ID = (E \oplus n_1) - IDP$.

STEP 5: *IDP and key updating*–The reader and the tag update the IDP and K as follows:

$$IDP^{(n+1)} = (IDP^{(n)} + (n_2 \oplus n_1)) \oplus ID,$$

$$K_1^{(n+1)} = K_1^{(n)} \oplus n_2 \oplus (K_3^{(n)} + ID),\ K_2^{(n+1)} = K_2^{(n)} \oplus n_2 \oplus (K_4^{(n)} + ID),$$

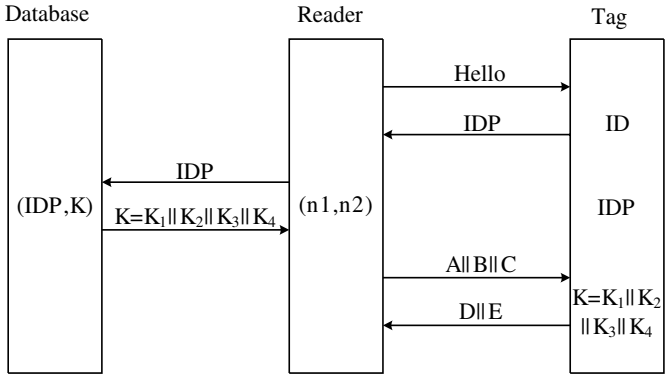


Fig. 1. The M^{2}AP and EMAP protocols. The reader interrogates the tag which responds with its current IDP. Using the tag's IDP, the reader looks up the database for the corresponding key K. The reader combines two random numbers (n_1,n_2), with the IDP and K to generate $A \parallel B \parallel C$ and authenticate itself to the tag. The tag responds with $D \parallel E$, where the tag's unique ID is embedded in E.

$$K_3^{(n+1)} = (K_3^{(n)} \oplus n_1) + (K_1^{(n)} \oplus ID),\ K_4^{(n+1)} = (K_4^{(n)} \oplus n_1) + (K_2^{(n)} \oplus ID).$$

The updated IDP and K are mutually stored for the next protocol run.

3.2 The EMAP Mutual Authentication Protocol

EMAP follows the same five steps described in M^{2}AP, with the only difference being the way that messages B, C, D, and E are generated and how the IDP and K are updated. In EMAP, the messages A, B, C, D, and E are generated as follows:

$$A = IDP \oplus K_1 \oplus n_1,\ B = (IDP \vee K_2) \oplus n_1,\ C = IDP \oplus K_3 \oplus n_2, \tag{3}$$

$$D = (IDP \wedge K_4) \oplus n_2,\ E = (IDP \wedge n_1 \vee n_2) \oplus ID \bigoplus_{i=1}^{4} K_i. \tag{4}$$

The updating of the IDP and K works as follows:

$$IDP^{(n+1)} = IDP^{(n)} \oplus n_2 \oplus K_1, \tag{5}$$
$$K_1^{(n+1)} = K_1^{(n)} \oplus n_2 \oplus (ID(1:48) \parallel F_p(K_4^{(n)}) \parallel F_p(K_3^{(n)})), \tag{6}$$
$$K_2^{(n+1)} = K_2^{(n)} \oplus n_2 \oplus (F_p(K_1^{(n)}) \parallel F_p(K_4^{(n)}) \parallel ID(49:96)), \tag{7}$$
$$K_3^{(n+1)} = K_3^{(n)} \oplus n_1 \oplus (ID(1:48) \parallel F_p(K_4^{(n)}) \parallel F_p(K_2^{(n)})), \tag{8}$$
$$K_4^{(n+1)} = K_4^{(n)} \oplus n_1 \oplus (F_p(K_3^{(n)}) \parallel F_p(K_1^{(n)}) \parallel ID(49:96)), \tag{9}$$

where $F_p(x)$ is the 24-bit sequence representing the parity of every 4 bit block of x.

4 Passive Attacks Against M²AP and EMAP

4.1 Passive Attack Against M²AP

In this section, we show how an adversary can extract the tag's unique ID by observing, on average, a *logarithmic* (in the length of ID) number of mutual authentication runs between the tag and the reader. To obtain the ID, the adversary needs to observe only the IDP, B, and E message exchange in each protocol run. For clarity, we first illustrate the attack via an example. Then, we show that the problem of extracting the tag's ID can be mapped to a set covering problem. Based on our mapping, we provide a probabilistic analysis of our attack.

Example: For clarity, we only show the values of the messages observed by the adversary that are relevant to the attack. Assume that the unique ID of an RFID tag is six bit long and is '001100'. Initially, the reader broadcasts a *"hello"* message to announce its presence. The tag challenges the reader by sending its current $IDP^{(1)} = 101100$. The reader looks up the database to find the corresponding secret key $K^{(1)}$, generates two random numbers ($n_1^{(1)}$,$n_2^{(1)}$), and challenges the tag with $A^{(1)} \parallel B^{(1)} \parallel C^{(1)}$, where $B^{(1)} = 011000$. After the reader is authenticated, the tag responds with $D^{(1)} \parallel E^{(1)}$, where $E^{(1)} = 101000$. Notice that, from message B in equation (1), if $(IDP)_i = 0$, then $(n_1)_i = (B)_i$, where the i subscript denotes the i^{th} bit of IDP, n_1, and B, respectively. Thus, the adversary can compute $n_1^{(1)} = *1**00$, where '$*$' represents an unknown bit. Therefore, $E^{(1)} \oplus n_1^{(1)} = *1**00$, and substituting into (2) we get:

$$ID = (E^{(1)} \oplus n_1^{(1)}) - IDP^{(1)} = *1**00 + 010100, \tag{10}$$

From the first protocol run, the adversary identifies the two least significant bits of the tag's ID as '00' using (10).

In the next protocol run, the adversary gathers the quantities $IDP^{(2)} = 010001$, $B^{(2)} = 111001$, and $E^{(2)} = 100100$. Using B in equation (1), the adversary computes $n_1^{(2)} = 1 * 100*$; hence, $E^{(2)} \oplus n_1^{(2)} = 0 * 110*$, and a second equation for the tag's ID can be constructed:

$$ID = (E^{(2)} \oplus n_1^{(2)}) - IDP^{(2)} = 0 * 110 * +101111. \tag{11}$$

Substituting the two least significant bits '00' in (11), the adversary can compute $ID = **1100$. By substituting '$**1100$' back in (10), the adversary identifies the fifth bit of the ID as '0', and substituting the five known bits of the ID back in (11), the adversary identifies the tag's unique ID as '001100'. Figure 2 presents the protocol exchanges for the two instances of mutual authentication between the tag and the reader in our example.

Let $x^{(n)} = E^{(n)} \oplus n_1^{(n)}$ denote the first term of the right hand side in (10) and let $(m)_i$ denote the i^{th} bit of message m. Note from our example that $(x^{(n)})_i$ is known if $(IDP^{(n)})_i = 0$. The set of equations, similar to (10) and (11), constructed by the adversary by observing protocol runs, can be solved

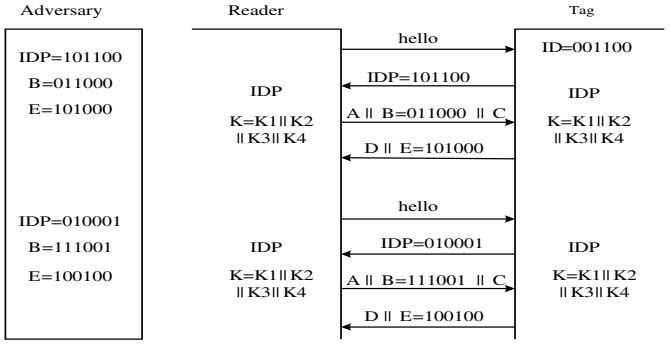


Fig. 2. Two mutual-authentication message exchanges between a reader and a tag. To the left is the information collected by an adversary that lead to disclosing the tag's unique ID

for the ID if $(x^{(n)})_i$ is known for *at least* one n. This condition is equivalent to $(IDP^{(n)})_i = 0$ for *at least* one n. That is, each position of the observed IDP's has a zero in *at least* one IDP. In turn, this observation can be *mapped* to a variant of the set cover problem expressed in the following lemma.

Lemma 1. *Let $S = \{1, 2, ..., L\}$, denote the indices of the L bits of the ID, and let $S_0^{(n)} = \{i \mid$ the i^{th} bit of the n^{th} IDP is $0\}$. The ID of the tag can be extracted if $\cup_n S_0^{(n)} = S$, that is, the union of $S_0^{(n)}$'s covers the entire set S.*

Proof. Let $\cup_n S_0^{(n)} = S$. Based on (2), we have

$$(ID) = ((E^{(n)}) \oplus (n_1^{(n)})) - (IDP^{(n)}) = x^{(n)} - (IDP^{(n)}). \tag{12}$$

For each bit of the ID we can write

$$(ID)_i = (x^{(n)})_i - (IDP^{(n)})_i + C_i^{(n)} \pmod 2, \tag{13}$$

where

$$C_i^{(n)} = f((x^{(n)})_{i-1}, (IDP^{(n)})_{i-1}, C_{i-1}^{(n)}) \tag{14}$$

denotes the carry from the modulo 2 addition in (12). To compute the carry C_i, we must know $(ID)_{i-1}$ from the LSB up to the $(i-1)^{th}$ bit. For the LSB, $C_1 = 0$ and hence, $(ID)_1$ can be extracted from any $S^{(n)}$ with $1 \in S^{(n)}$. Once $(ID)_1$ has been extracted, using (13), $(x^{(n)})_1$ can be extracted for all n. Therefore, using (14), C_2 can be extracted for all n. Now, equation (13), can be used to solve for $(ID)_2$ from any $S^{(n)}$ with $2 \in S^{(n)}$. Since for all $i \in [1 : L]$, there exists an $S^{(n)}$ with $i \in S^{(n)}$, one can recursively solve for the tag's ID from the least significant to the most significant bits.

Given that the bit values of the IDP are drawn from a probability distribution, we can compute the average number of protocol runs required to recover the unique ID using the following lemma.

Table 1. Mapping the ID recovery problem to a set cover problem

ID **recovery problem**	$\leftrightarrow$	**Set covering problem**
$S = \{1, 2, ..., L\}$	$\leftrightarrow$	Entire set S
$IDP^{(n)}$ containing k zeros	$\leftrightarrow$	Subset of S with cardinality k
Observing protocol runs with IDP's having at least one zero in every position	$\leftrightarrow$	Finding a set of subsets of S that covers S

Lemma 2. *Let p be the probability of any bit of the IDP being 1, and let L be the length of the IDP. Then, the probability of fully disclosing the tag's unique ID by observing m mutual authentication runs, is given by:*

$$\Pr(\textit{disclosing the } ID \textit{ after } m \textit{ messages}) = (1 - p^m)^L. \tag{15}$$

Moreover, given any $\epsilon \in (0, 1)$, the number of mutual authentication runs an eavesdropper has to observe in order to extract the ID with probability at least $1 - \epsilon$ is given by:

$$m = \lceil \frac{\ln(1 - \exp^{\frac{\ln(1-\epsilon)}{L}})}{\ln p} \rceil. \tag{16}$$

Proof. The proof of lemma 2 is provided in the appendix.

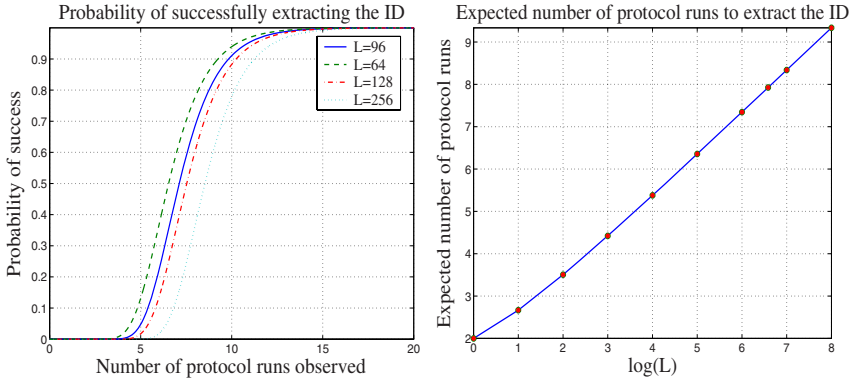
From lemma 2, we observe that the probability of extracting the ID of the tag is a monotonically increasing function of the number of observed protocol runs m, converging to 1. Lemma 2 also specifies how many protocol runs one must eavesdrop, before the entire ID can be extracted, with a desired probability. In Figure 3(a), we show the probability of extracting the ID of the tag as a function of the number of protocol runs observed, for different values of L and for $p = \frac{1}{2}$. We now compute the average number of protocol runs an adversary needs to eavesdrops to extract the tag's ID.

Lemma 3. *Let p be the probability of any bit of the IDP being 1, and let L be the length of the IDP. Let m denote the number of protocol runs needed to extract the tag's unique ID. Then, the expected value of m is:*

$$E[m] \ = \ \sum_{k=1}^{L} \binom{L}{k} \frac{(-1)^{k+1}}{1 - p^k}. \tag{17}$$

Proof. The proof of lemma 3 is provided in the appendix.

For $L = 96$ and $p = \frac{1}{2}$ as specified in [1], the expected number of protocol runs needed to extract the tag's unique ID is 7.9252. Figure 3 (b) shows the analytically derived relation between the expected number of protocol runs needed to extract the tag's ID and the length of the ID. We observe that $E[m]$ grows linearly with the logarithm of the ID length L.



(a) (b)

Fig. 3. (a) The probability of extracting the ID of the tag as a function of the number of protocol runs observed, for varying ID lengths, (b) The expected number of messages needed to extract the ID as a function of the length of the ID in a logarithmic scale

4.2 Passive Attack Against EMAP

In EMAP [2], a particular emphasis is put on the properties of message E in equation (4), due to the fact that the tag's unique ID is extracted via E [2]. However, the ID is also used in equations (6) - (9) for the key updating. Therefore, an adversary can extract the tag's ID using equations (6) - (9), without breaking message E. Let S_0 denote the set of indexes where the bits of the IDP are 0 and S_1 denote the set of indexes where the bits of the IDP are 1, that is,

$$S_0 = \{i \mid the\ i^{th}\ bit\ of\ IDP\ is\ 0\}, \tag{18}$$

$$S_1 = \{i \mid the\ i^{th}\ bit\ of\ IDP\ is\ 1\}. \tag{19}$$

The attack against EMAP consists of the following steps:

Step 1: From message B in equation (3), we have $(IDP)_i \vee (K_2)_i = 1,\ \forall i \in S_1$, regardless of the values of $(K_2)_i$. Therefore, $(n_1)_i = \overline{(B)_i},\ \forall i \in S_1$, where $\overline{b}$ denotes the complement of the bit b.

Step 2: Message A in equation (3) has two unknowns, namely, the secret key K_1 and the nonce n_1. The partial information about n_1 obtained from Step 1 can be substituted in (3) to extract bits of K_1,

$$(K_1)_i = (A)_i \oplus (n_1)_i \oplus (IDP)_i, \quad \forall\ i \in S_1. \tag{20}$$

Step 3: From the message D in equation (4), we have $(IDP)_i \wedge (K_4)_i = 0,\ \forall i \in S_0$ regardless of the values $(K_4)_i$. Therefore, $(n_2)_i = (D)_i,\ \forall i \in S_0$.

Step 4: Equation (5) has two unknowns, the secret key K_1 and the nonce n_2. The partial information about n_2 obtained in Step 3 can be substituted in (5) to extract bits of K_1 as follows:

$$(K_1)_i = (IDP^{(n+1)})_i \oplus (IDP^{(n)})_i \oplus (n_2)_i, \quad \forall\ i \in S_0. \tag{21}$$

Up to this point, the adversary knows $(K_1)_i,\ \forall i \in S_1$ from (20), and $(K_1)_i,\ \forall i \in S_0$ from (21). Hence, the secret key K_1 has been fully extracted.

Step 5: By substituting K_1 into (3) and (5), the adversary obtains n_1 and n_2 as shown below:

$$n_1 = A \oplus IDP^{(n)} \oplus K_1^{(n)},\ n_2 = IDP^{(n+1)} \oplus IDP^{(n)} \oplus K_1^{(n)}. \tag{22}$$

Step 6: By eavesdropping the next protocol run, the adversary can extract the updated value $K_1^{(n+1)}$ as described in Steps 1-4. Substituting the values of $K_1^{(n)}$ and $K_1^{(n+1)}$ in (6), the first half of the tag's unique ID is revealed:

$$(ID)_i = (K_1^{(n+1)})_i \oplus (K_1^{(n)})_i \oplus n_2, \quad \forall i \in [1 : \frac{L}{2}]. \tag{23}$$

Step 7: From messages B and D in equations (3) and (4) respectively, $(K_2)_i = (B)_i \oplus (n_1)_i,\ \forall i \in S_0$ and $(K_4)_i = (D)_i \oplus (n_2)_i,\ \forall i \in S_1$. Therefore, in every protocol run, the bits of K_2 corresponding to the zero bits of IDP are known, and the bits of K_4 corresponding to the one bits of IDP are known. Thus, for $i = \frac{L}{2} + 1 : L$, if $(IDP^{(n)})_i = (IDP^{(n+1)})_i = 0$ then $(K_2^{(n)})_i$ and $(K_2^{(n+1)})_i$ are known and, hence, using equation (7),

$$(ID)_i = (K_2^{(n)})_i \oplus (K_2^{(n+1)})_i \oplus (n_2)_i. \tag{24}$$

Likewise, if $(IDP^{(n)})_i = (IDP^{(n+1)})_i = 1$ then $(K_4^{(n)})_i$ and $(K_4^{(n+1)})_i$ are known, and using equation (9),

$$(ID)_i = (K_4^{(n)})_i \oplus (K_4^{(n+1)})_i \oplus (n_1)_i. \tag{25}$$

Using (24) and (25), the second half bits of the ID are extracted if two consecutive IDP's have the same bit value in that position. Hence, the adversary can solve for each bit in the second half of the ID depending on the value of the IDP's in its position. Lemma 4 analyzes the performance of our passive attack against EMAP.

Lemma 4. *Let p be the probability of any bit of the IDP being 1, and let L be the length of the IDP. Then, the probability of fully disclosing the tag's unique ID by observing m consecutive mutual authentication runs, is given by:*

$$Pr(\textit{extracting the ID after } m \textit{ messages}) = \begin{cases} 0, & m < 2 \\ (1 - (2p - 2p^2)^m)^{\frac{L}{2}}, & m \geq 2 \end{cases} \tag{26}$$

Moreover, given any $\epsilon \in (0, 1)$, the number of consecutive mutual authentication runs an eavesdropper has to observe to extract the entire static ID with probability at least $1 - \epsilon$ is given by:

$$m = \lceil 1 + \frac{\ln(1 - \exp^{\frac{2\ln(1-\epsilon)}{L}})}{\ln(2p - 2p^2)} \rceil. \tag{27}$$

Furthermore, the average number of consecutive mutual authentication runs needed to extract the tag's unique ID is given by:

$$E[m] \;=\; 1+\sum_{k=1}^{\frac{L}{2}} \binom{\frac{L}{2}}{k} \frac{(-1)^{k+1}}{1-(2p-2p^2)^k}. \tag{28}$$

Proof. In our attack, no information about the ID can be extracted by just eavesdropping the first protocol run. However, by eavesdropping two consecutive protocol runs, the adversary is guaranteed to recover the first half of the ID (Steps 1-6). For the second half of the ID, bits are recovered probabilistically by solving the update equation of K_2 or K_4 (Step 7). When two consecutive IDP's have the same value in one bit position, the adversary can solve for the bit of the ID at that position. That is, for the i^{th} bit of ID, the adversary will successfully solve for its value if $(IDP^{(n)})_i = (IDP^{(n+1)})_i$, *i.e.* both are 0 or 1 which occur with probabilities $(1-p)^2$ and p^2, respectively. This means that the probability of successfully solving for each bit of the second half of the ID is given by $(1-p)^2+p^2$. Also note that, to extract all bits of the second half of the ID, a match between two consecutive IDPs has to occur, $\forall i \in [\frac{L}{2}+1 : L]$. Hence, the problem of extracting all second half ID bits, can be mapped to the same set covering problem expressed in Lemma 1, with a different success probability, and ID length equal to $\frac{L}{2}$. Following the same analysis as in Lemma 2 and 3, we can compute the quantities in (26), (27), and (28) by substituting the success probability for M^{2}AP '$(1-p)$', with $(1-p)^2+p^2$. Note that at least two protocol runs are needed to extract useful information and, hence, 1 is added to the expressions in (27), and (28).

In Figure 4(a),(b), we show the histogram of the probability of extracting the tag ID after eavesdropping exactly m protocol runs for M^{2}AP and EMAP, respectively, for $L = 96$ and $p = 0.5$. Note that the average number of protocol runs required to extract the tag ID is 7.9273 and 7.9223 for M^{2}AP and EMAP respectively, while the theoretical result obtained by Lemma 2 shows that $E[m] = 7.9252$.

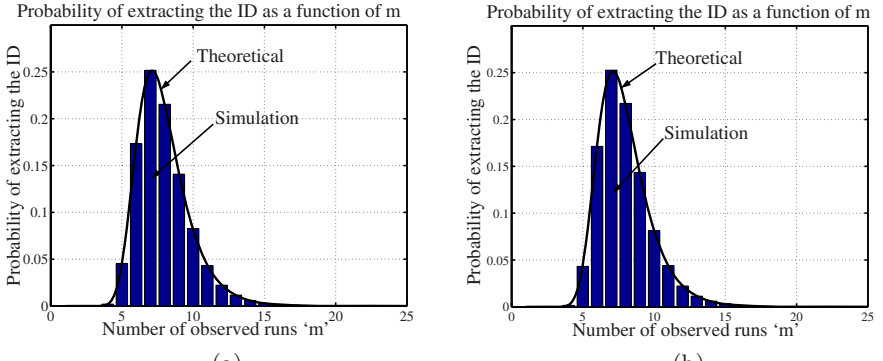


Fig. 4. Probability of extracting the tag ID after eavesdropping exactly m protocol runs when $L = 96$ and $p = 0.5$ for (a) the M^{2}AP protocol, (b) the EMAP protocol

5 Related Work

The problem of mutual authentication in RFID systems has been studied under different constraints [1,2,6,9,10,12,14]. Juels and Pappu have suggested the use of a public key cryptosystem to solve the problem of consumer privacy in RFID banknotes [9]. Avoine, however, described possible limitations on the security of the protocol in [11]. Feldhofer et.al. proposed the use of AES symmetric key cipher to achieve mutual authentication between tags and readers for tags able to perform AES Encryption/Decryption [10]. Weis et.al. described privacy and security risks for RFID systems and proposed solutions based on one way hash functions [12]. Juels proposed the use of a pool of pseudonyms for each tag to protect the privacy of the tag's ID [8].

Vajda and Buttyan proposed lightweight cryptographic primitives for tag authentication based on simple bitwise operations [14]. In [6], Juels and Weis proposed HB$^+$, a lightweight authentication protocol based on the human-to-computer authentication protocol designed by Hopper and Blum [5]. The security proof of the HB$^+$ against active attacks was based on the Learning Parity with Noise (LPN) problem. Gilbert et.al., however, showed a linear time active attack on the HB$^+$ protocol [7].

In [1,2], Peris-Lopez et.al. proposed M^{2}AP and EMAP, mutual authentication protocols that we analyze in this paper. In [3], Li and Wang describe active attacks against M^{2}AP that require $O(L)$ interactions between the adversary and the tag to extract its ID. Our attack requires passive observation of an average of $O(\log_2 L)$ protocol runs to extract the tag's ID. Li and Deng described active attacks against EMAP in [4]. Their attack relies on active probing of the tag via rogue reader.

6 Conclusion

In this paper, we addressed the problem of mutual authentication in RFID systems. We analyzed M^{2}AP and EMAP, two lightweight mutual authentication protocols and showed how a passive adversary can extract the tag's unique ID by observing, on average, a logarithmic (in the length of the ID) number of protocol runs. We provided a probabilistic analysis of our attacks by mapping the problem of extracting the tag's ID to a set covering problem.

References

1. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J.M., Ribagorda, A.: M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, Springer, Heidelberg (2006)
2. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J.M., Ribagorda, A.: EMAP: An Efficient Mutual Authentication Protocol for Low-cost RFID Tags. In: OTM Federated Conferences and Workshop: IS Workshop. LNCS, Springer, Heidelberg (2006)

3. Li, T., Wang, G.: Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols. In: IFIP SEC (2007)
4. Li, T., Deng, R.H.: Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol. In: AReS 2007: Second International Conference on Availability, Reliability and Security (2007)
5. Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, Springer, Heidelberg (2001)
6. Juels, A., Weis, S.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, Springer, Heidelberg (2005)
7. Gilbert, H., Robshaw, M., Sibert, H.: An Active Attack Against HB^+ – A provably Secure Lightweight Authentication Protocol Protocol (2005)
8. Juels, A.: Minimalist Cryptography for Low-Cost RFID Tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, Springer, Heidelberg (2005)
9. Juels, A., Pappu, R.: Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, Springer, Heidelberg (2003)
10. Feldhofer, M., Aigner, M., Dominikus, S.: An Application of RFID Tags using Secure Symmetric Authentication. In: SecPerU 2005. International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (2005)
11. Avoine, G.: Privacy Issues in RFID Banknote Protection Schemes. In: International Conference on Smart Card Research and Advanced Applications – CARDIS (2004)
12. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: SPC 2003. International Conference on Security in Pervasive Computing (2003)
13. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient Hash-Chain Based RFID Privacy Protection Scheme. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, Springer, Heidelberg (2004)
14. Vajda, I., Buttyán, L.: Lightweight Authentication Protocols for Low-Cost RFID Tags. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, Springer, Heidelberg (2003)
15. Defend, B., Fu, K., Juels, A.: Cryptanalysis of Two Lightweight RFID Authentication Schemes. In: International Workshop on Pervasive Computing and Communication Security – PerSec (2007)
16. Juels, A.: RFID Security and Privacy: A research Survey (2005)
17. Avoine, G.: Bibliography on Security and Privacy in RFID Systems `http://lasecwww.epfl.ch/gavoine/rfid/`
18. Buchmann, J.A.: Introduction to cryptography. Springer, Heidelberg (2004)
19. Garfinkel, S.L., Juels, A., Pappu, R.: RFID Privacy: An overview of Problems and Proposed Solutions. IEEE Security & Privacy (2005)
20. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems using the AES Algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, Springer, Heidelberg (2004)

Appendix

Let $(M^{(i)})_j$ denote the j^{th} bit of the i^{th} message. Define $S^{(m)} = \bigcap_{k=1}^{m} M^{(k)}$ to be the result of bitwise AND for the messages $M^{(1)}$ through $M^{(m)}$, and define the random variable X_i as follows:

$$X_i = \begin{cases} 1, & \sum_{j=1}^{L} (S^{(i)})_j \;=\; 0, \\ 0, & otherwise. \end{cases} \tag{29}$$

Then, if the probability of any bit of $M^{(i)}$ being 1 is equal to p, we get

$$\Pr((S^{(m)})_k = 1) = \Pr(\bigcap_{l=1}^{m}\{M_k^{(l)} = 1\}) = \prod_{l=1}^{m}\Pr\{M_k^{(l)} = 1\} = p^m, \tag{30}$$

$$Pr(S_k^{(m)} = 0) = 1 - Pr(S_k^{(m)} = 1) = 1 - p^m. \tag{31}$$

Proof of Lemma 2

From equation (31) and by independence of bits, we get:

$$\Pr(X_m = 1) = (1 - p^m)^L. \tag{32}$$

Therefore, for any $\epsilon > 0$, we have:

$$\begin{aligned}
\Pr(X_m = 1) &> 1 - \epsilon \\
&\Rightarrow (1 - p^m)^L > 1 - \epsilon \\
&\Rightarrow 1 - p^m > \exp^{\frac{\ln(1-\epsilon)}{L}} \\
&\Rightarrow p^m < 1 - \exp^{\frac{\ln(1-\epsilon)}{L}} \\
&\Rightarrow m > \frac{\ln(1 - \exp^{\frac{\ln(1-\epsilon)}{L}})}{\ln p}.
\end{aligned} \tag{33}$$

Proof of Lemma 3

Define the random variable Y to be the number of messages such that $X_m = 1$ for the first time, then Y can be written as $Y = \min_i\{X_i = 1\}$. Then, $\{Y = i\} \Leftrightarrow \{X_i = 1 \; and \; X_{i-1} = 0\}$ and, hence,

$$\begin{aligned}
\Pr(Y = i) &= \Pr(X_i = 1, X_{i-1} = 0) = \Pr(X_i = 1 \mid X_{i-1})\Pr(X_{i-1}) \\
&= \sum_{j=0}^{L-1}\Pr(X_i = 1 \mid \sum_{k=1}^{L} S_k^{(i-1)} = j)\Pr(\sum_{k=1}^{L} S_k^{(i-1)} = j),
\end{aligned} \tag{34}$$

where in equation (34) we sum over all possible number of zeros in $S^{(i-1)}$. But,

$$\Pr(X_i = 1 \mid \sum_{k=1}^{L} S_k^{(i-1)} = j) = (1 - p)^{L-j} \tag{35}$$

because for all bits where $S_k^{(i-1)} = 1$, $M_k^{(i)} = 0$ to satisfy $X_i = 1$. Equation (34) can be shown as follows: From (30), the probability of a single bit in $S^{(i-1)}$ to be equal to 1 is p^{i-1}. Therefore, by independence, the probability of having $L - j$ ones in $S^{(i-1)}$ is given by:

$$\Pr(\sum_{k=1}^{L} S_k^{(i-1)} = L - j) = (p^{i-1})^{L-j}. \tag{36}$$

Similarly, from (31), the probability of any single bit of $S^{(i-1)}$ to be equal to 0 is $1-p^{i-1}$ and, hence,

$$\Pr(S^{(i-1)}\ has\ j\ zeros) = (1-p^{i-1})^j. \tag{37}$$

Finally, there are $\binom{L}{j}$ different ways of choosing the positions of the j zeros. Thus, by combining (36) and (37), the probability of having exactly j zeros in $S^{(i-1)}$ can be written as:

$$\Pr(\sum_{k=1}^{L} S_k^{i-1} = j) = \binom{L}{j}(1-p^i)^j (p^i)^{L-j}. \tag{38}$$

From (35), (38), and $1-p^{i-1} = (1-p)\sum_{k=0}^{i-2} p^k$, it follows that,

$$\begin{aligned}
\Pr(Y=i) &= \sum_{j=0}^{L-1} (1-p)^{L-j}\ \binom{L}{j}\ (1-p^{i-1})^j\ (p^{i-1})^{L-j} \\
&= (1-p)^L \sum_{j=0}^{L-1} \binom{L}{j} (\sum_{k=0}^{i-2} p^k)^j\ (p^{i-1})^{L-j} \\
&= (1-p)^L\ [(\sum_{k=0}^{i-1} p^k)^L\ -\ (\sum_{k=0}^{i-2} p^k)^L] = (1-p^i)^L - (1-p^{i-1})^L.
\end{aligned}$$

Hence, the expected number of messages required is:

$$\begin{aligned}
E[Y] &= \sum_{i=1}^{\infty} i\ \Pr(Y=i) = \sum_{i=1}^{\infty} i\ [(1-p^i)^L - (1-p^{i-1})^L] \\
&= \sum_{i=1}^{\infty} i\ [\sum_{k=0}^{L} \binom{L}{k}(-1)^k (p^k)^i - \sum_{k=0}^{L} \binom{L}{k}(-1)^k (p^k)^{i-1}] \\
&= \sum_{i=1}^{\infty} i\ \sum_{k=0}^{L} (p^k-1)\binom{L}{k}(-1)^k (p^k)^{i-1} = \sum_{k=1}^{L} (p^k-1)\binom{L}{k}(-1)^k \sum_{i=1}^{\infty} i\ (p^k)^{i-1} \\
&= \sum_{k=1}^{L} (p^k-1)\binom{L}{k}(-1)^k (\frac{1}{1-p^k})^2 = \sum_{k=1}^{L} \binom{L}{k}(-1)^{k+1} (\frac{1}{1-p^k})
\end{aligned}$$

and the lemma follows.