Passive Attack Against the M^{2}AP Mutual Authentication Protocol for RFID Tags *

Mihály Bárász
ELTECRYPT Research Group,
Eötvös University
Budapest, Hungary
klao@cs.elte.hu

Balázs Boros
ELTECRYPT Research Group,
Eötvös University
Budapest, Hungary
borbal@cs.elte.hu

Péter Ligeti
ELTECRYPT Research Group,
Eötvös University;
Alfréd Rényi Institute of Mathematics
Budapest, Hungary
turul@cs.elte.hu

Krisztina Lója
ELTECRYPT Research Group,
Eötvös University;
University of Technology and Economics
Budapest, Hungary
loja@math.bme.hu

Dániel A. Nagy
ELTECRYPT Research Group,
Eötvös University
Budapest, Hungary
nagydani@epointsystem.org

***Abstract* - In this paper, we present a passive attack for finding out the secrets used in M^{2}AP (Minimalist Mutual Authentication Protocol), which is an authentication protocol between RFID tags and RFID readers. We describe an algorithm that breaks the protocol after eavesdropping a few consecutive rounds of communication. After two eavesdropped runs of the protocol, the attacker can learn the identification number of the tag and some of the common secrets shared by the tag and the reader. For finding out all of the secrets, the attacker needs to eavesdrop some more rounds of the protocol. This means that in the subsequent rounds the attacker can successfully impersonate both the targeted tag and the reader.**

***Keywords* - RFID, Tag, Reader, Mutual Authentication, M^{2}AP, Passive Attack**

I. Introduction

In a mutual authentication protocol for RFID applications, the goal is to prevent unauthorized readers from reading some or all information stored in the RFID tags, while providing authorized readers with the capability of distinguishing between authorized and unauthorized tags. The security of such a protocol depends on the costs that it imposes on potential attackers who might want to impersonate the either tags or the readers without being authorized to do so.

In the particular case of M^{2}AP, Peris-Lopez et al. [1] propose a protocol in which authorization is provided by a common secret shared by authorized readers and tags. The goal of reader authentication is to prevent unauthorized readers from reading the identification number of authorized tags. Note that for some applications this is not sufficient: in military applications, authorized tags must not respond to unauthorized readers at all.

M^{2}AP sets forth the very attractive, but also very challenging goal of low complexity in tags while maintaining adequate levels of security. It refrains from using traditional cryptographic primitives, doing just elementary arithmetics in tags. While we do not claim that this ambitious goal cannot be achieved, in this paper we demonstrate that M^{2}AP certainly falls short of achieving it.

M^{2}AP is a slight improvement over LMAP (Lightweight Mutual Authentication Protocol) proposed by Peris-Lopez et al. [2], which similarly uses elementary bitwise operations. LMAP can be broken slightly easier than M^{2}AP. For a passive attack against LMAP see Bárász et al. [3].

Recently, Li and Wang [4] have pointed out some weaknesses of M^{2}AP and LMAP as well. The authors show two kinds of active attacks against the protocol. The first one is able to de-synchronize the communication between the tag and the reader, the second one is a man-in-the-middle attack which is able to get the whole secret key of the tag after a de-synchronization phase. The authors detail their attack against LMAP and go on to claim that the attack can be adapted for M^{2}AP easily. We will see that this is not at all trivial to find out two of the four secret keys used in the protocol. It requires more work than in the case of LMAP.

Our passive attack is passive in the sense that it uses only eavesdropping, so we do not have to take any technical assumption about the protocol, carrying out the attack is possible under the correct working of the protocol. A MATLAB implementation of our attack can be found at the homepage of ELTECRYPT Research Group [5].

In this paper, we show a fully passive attack against the protocol, which is able to get every secret information stored in the tag only by eavesdropping a few consecutive rounds of the communication. The rest of this paper is organized as follows. In Section II., we introduce the M^{2}AP protocol of Peris-Lopez et al [1]. In Section III., we point out the main weaknesses of the protocol. In Section IV., we present our passive attack step by step. In Section V., we review the possible attacks and finally, in Section VI., we give some general remarks on security of protocols.

II. The M^{2}AP Protocol

We give a brief introduction to M^{2}AP. For more details please refer to [1]. Each tag has a unique identification number (ID) that never changes. Also, each tag has an index-pseudonym (IDS) and four secret keys (K_1, K_2, K_3 and K_4) that must be updated after every authentication round. Before each authentication, the reader generates two random numbers (n_1 and n_2). We will consider only the case of one tag. The protocol uses bitwise XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$) and addition modulo 2^{96} ($+$).

$K_1, K_2, K_3, K_4, ID, IDS, n_1, n_2$ are vectors of 96 bits. The nth round of the protocol consists of the following four steps (the upper index between the parentheses denotes the number of the present round):

1. Tag Identification

$$\boxed{\text{Reader}} \rightarrow hello \rightarrow \boxed{\text{Tag}}$$

$$\boxed{\text{Reader}} \leftarrow IDS^{(n)} \leftarrow \boxed{\text{Tag}}$$

After receiving a $hello$ message from the reader, the tag sends its actual IDS to the reader. By means of IDS, the reader will be able to

*This research was partially supported by Jedlik Ányos Program NKFP2-00027/2005, and MIK Grant

access the tag's secret keys (K_1, K_2, K_3 and K_4). Furthermore, the reader is also able to access the tag's ID.

2. Reader Authentication

$$\boxed{\text{Reader}} \rightarrow A^{(n)} := IDS^{(n)} \oplus K_1^{(n)} \oplus n_1^{(n)} \rightarrow \boxed{\text{Tag}}$$

$$\boxed{\text{Reader}} \rightarrow B^{(n)} := (IDS^{(n)} \wedge K_2^{(n)}) \vee n_1^{(n)} \rightarrow \boxed{\text{Tag}}$$

$$\boxed{\text{Reader}} \rightarrow C^{(n)} := IDS^{(n)} + K_3^{(n)} + n_2^{(n)} \rightarrow \boxed{\text{Tag}}$$

From message A, the tag can calculate the random value denoted by n_1. Knowing n_1, the tag can also calculate message B and if it is the same as message B received from the reader, the tag establishes that the reader knows K_1 and K_2. Thus, the authentication of the reader is ready. From message C, the tag can calculate the random number n_2.

3. Tag Authentication

$$\boxed{\text{Reader}} \leftarrow D^{(n)} := (IDS^{(n)} \vee K_4^{(n)}) \wedge n_2^{(n)} \leftarrow \boxed{\text{Tag}}$$

$$\boxed{\text{Reader}} \leftarrow E^{(n)} := (IDS^{(n)} + ID) \oplus n_1^{(n)} \leftarrow \boxed{\text{Tag}}$$

Once these verifications are performed, the tag will generate the answer message D and E to authenticate and transmit its static identifier in a secure way.

4. Updating the Values of IDS, K_1, K_2, K_3 and K_4

$$\begin{aligned}
IDS^{(n+1)} &:= \left(IDS^{(n)} + (n_1^{(n)} \oplus n_2^{(n)})\right) \oplus ID, \\
K_1^{(n+1)} &:= K_1^{(n)} \oplus n_2^{(n)} \oplus (K_3^{(n)} + ID), \\
K_2^{(n+1)} &:= K_2^{(n)} \oplus n_2^{(n)} \oplus (K_4^{(n)} + ID), \\
K_3^{(n+1)} &:= (K_3^{(n)} \oplus n_1^{(n)}) + (K_1^{(n)} \oplus ID), \\
K_4^{(n+1)} &:= (K_4^{(n)} \oplus n_1^{(n)}) + (K_2^{(n)} \oplus ID).
\end{aligned}$$

After a successful mutual authentication, the tag and the reader also update the index-pseudonym and the four secret keys.

III. Weaknesses

Every bit affects only the bits which are to the left from that given bit. Hence, each bit depends only on bits with the same or bigger indices. In particular, the least significant bits are independent of every other bit. This is so because M^{2}AP uses only bitwise operations and addition modulo 2^{96}.

Taking into account only the least significant bits, XOR operation and addition modulo 2^{96} are the same. We can use this observation to deduce the least significant bits. In 4.1.1 and 4.2.1 we do not take difference between $\oplus$ and $+$ (and we use the notation $\oplus$).

The bitwise OR and AND operations in messages B and D are the other weak points of the protocol. From messages B and D, one can easily gain information about the random numbers n_1 and n_2 with the help of the set and reset bits of IDS, respectively.

The addition modulo 2^{96} poses no difficulty if we know every bit on the right hand side.

IV. Steps of Finding Out Secrets

In this section, we present our solution for finding out the secret identification number and the secret keys.

Let us denote the kth bit of M in round n by $[M^{(n)}]_k$ for $M \in \{A, B, C, D, E, K_1, K_2, K_3, K_4, IDS, n_1, n_2\}$. For example, $[K_1^{(n)}]_{96}$ is the least significant bit of key K_1 in round n. $[ID]_k$ will mean the kth bit of ID in any of the rounds, since this is a constant sequence of bits.

Since every information is communicated via an insecure public radio channel, after round n, $IDS^{(n)}$ and messages $A^{(n)}, B^{(n)}, C^{(n)}, D^{(n)}, E^{(n)}$ are known to the attacker eavesdropping the communication between the tag and the reader. We will denote the bits just obtained by underlining them (e.g. $\underline{[n_1^{(n)}]_k}$).

An attacker can get the secrets stored in the tag in two steps: the identification number ID of the tag , two of the secret keys K_1 and K_3 and the random numbers n_1 and n_2 can be find out by case-by-case analysis and simple arithmetic operations on the eavesdropped data, to get the remaining two secret keys K_2 and K_4 we need some other techniques.

4.1 Finding out K_1, K_3, n_1, n_2 and ID

In this subsection we show the following: if the attacker can eavesdrop two consecutive rounds of the protocol of the same tag, i.e. she knows $IDS^{(n)}$, $A^{(n)}$, $B^{(n)}$, $C^{(n)}$, $D^{(n)}$, $E^{(n)}$, $IDS^{(n+1)}$, $A^{(n+1)}$, $B^{(n+1)}$, $C^{(n+1)}$, $D^{(n+1)}$ and $E^{(n+1)}$, then she can calculate easily $n_1^{(n)}$, $n_2^{(n)}$, $K_1^{(n)}$, $K_3^{(n)}$, $n_1^{(n+1)}$, $n_2^{(n+1)}$, $K_1^{(n+1)}$, $K_3^{(n+1)}$, $K_1^{(n+2)}$, $K_3^{(n+2)}$, $IDS^{(n+2)}$ and ID.

4.1.1 The Least Significant Bits of K_1, K_3, n_1, n_2 and ID

The attack will use only two consecutive authentications for finding out the least significant bits of K_1, K_3, n_1, n_2 and ID, so let us suppose that the attacker knows the following bits only: $[IDS^{(n)}]_{96}$, $[A^{(n)}]_{96}$, $[B^{(n)}]_{96}$, $[C^{(n)}]_{96}$, $[D^{(n)}]_{96}$, $[E^{(n)}]_{96}$, $[IDS^{(n+1)}]_{96}$, $[A^{(n+1)}]_{96}$, $[B^{(n+1)}]_{96}$, $[C^{(n+1)}]_{96}$, $[D^{(n+1)}]_{96}$ and $[E^{(n+1)}]_{96}$. First, the attacker can calculate $\underline{[n_2^{(n)}]_{96}}$:

$$[n_2^{(n)}]_{96} = [E^{(n)}]_{96} \oplus [IDS^{(n+1)}]_{96}.$$

Then she can obtain $\underline{[K_3^{(n)}]_{96}}$ from message C:

$$[K_3^{(n)}]_{96} = [C^{(n)}]_{96} \oplus [IDS^{(n)}]_{96} \oplus [n_2^{(n)}]_{96}.$$

The next bit what she can calculate is $\underline{[ID]_{96}}$:

$$\begin{aligned}
&[ID]_{96} = \\
&= ([K_1^{(n)}]_{96} \oplus [ID]_{96}) \oplus ([K_1^{(n+1)}]_{96} \oplus [ID]_{96}) \oplus \\
&\oplus [K_3^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} = \\
&= ([A^{(n)}]_{96} \oplus [E^{(n)}]_{96}) \oplus ([A^{(n+1)}]_{96} \oplus [E^{(n+1)}]_{96}) \oplus \\
&\oplus [K_3^{(n)}]_{96} \oplus [n_2^{(n)}]_{96},
\end{aligned}$$

where the first equality follows from the updating formula of $K_1^{(n+1)}$.

After the attacker has found out ID, she can obtain $[n_1^{(n)}]_{96}$ and $[K_1^{(n)}]_{96}$ similarly as above.

First, the attacker can calculate $\underline{[n_1^{(n)}]_{96}}$ with the help of the definition of $E^{(n)}$:

$$[n_1^{(n)}]_{96} = [E^{(n)}]_{96} \oplus [IDS^{(n)}]_{96} \oplus [ID]_{96}.$$

Finally, she can get $\underline{[K_1^{(n)}]_{96}}$ from the definition of $A^{(n)}$:

$$[K_1^{(n)}]_{96} = [A^{(n)}]_{96} \oplus [IDS^{(n)}]_{96} \oplus [n_1^{(n)}]_{96}.$$

So far we have shown how the attacker can find out $[n_1^{(n)}]_{96}$, $[n_2^{(n)}]_{96}$, $[K_1^{(n)}]_{96}$, $[K_3^{(n)}]_{96}$ and $[ID]_{96}$. After that she can calculate easily $\underline{[K_1^{(n+1)}]_{96}}$ and $\underline{[K_3^{(n+1)}]_{96}}$ with the help of their updating formulas, $\underline{[n_1^{(n+1)}]_{96}}$ from $[E^{(n+1)}]_{96}$, and with these bits, she can get $\underline{[n_2^{(n+1)}]_{96}}$ also on an easy way. After the attacker has computed these bits, she can get $\underline{[IDS^{(n+2)}]_{96}}$, $\underline{[K_1^{(n+2)}]_{96}}$ and $\underline{[K_3^{(n+2)}]_{96}}$ also from the updating formulas.

4.1.2 The Bits Immediately Before the Least Significant Ones in K_1, K_3, n_1, n_2 and ID

The only thing an attacker needs to do is setting up the adequate equations applied in 4.1.1. The main point is that the attacker can handle addition modulo 2^{96} for the 95th bits if she knows the 96th bits of the addends. For example, if $[K_3^{(n)}]_{96} \wedge [ID]_{96} = 1$ then updating formula of $[K_1^{(n+1)}]_{95}$ gets the following form:

$$[K_1^{(n+1)}]_{95} = [K_1^{(n)}]_{95} \oplus [n_2^{(n)}]_{95} \oplus [K_3^{(n)}]_{95} \oplus [ID]_{95} \oplus 1.$$

4.1.3 The More Significant Bits of K_1, K_3, n_1, n_2 and ID

It is clear that the attacker can derive all the bits of K_1, K_3, n_1, n_2 and ID using the methods described in 4.1.1 and 4.1.2. Here we can see the main weakness of M^{2}AP, i.e. we can get the only static secret identification number ID after eavesdropping only two consecutive rounds of communications.

4.2 Finding out K_2 and K_4

So far the attacker has not used messages B and D. These are containing information about K_2 and K_4. Namely, the following implications are hold for $n \leq i$ and $1 \leq k \leq 96$:

$$([IDS^{(i)}]_k = 1) \wedge ([n_1^{(i)}]_k = 0) \Rightarrow [B^{(i)}]_k = [K_2^{(i)}]_k,$$

$$([IDS^{(i)}]_k = 0) \wedge ([n_2^{(i)}]_k = 1) \Rightarrow [D^{(i)}]_k = [K_4^{(i)}]_k.$$

This means that the attacker learns about one fourth of keys K_2 and K_4 in each round of the protocol. With this information she can derive the whole of K_2 and K_4 after eavesdropping a few consecutive rounds of the protocol. This can be done in three different ways.

4.2.1 First Way

First, we show how the attacker can obtain $K_2^{(n+192)}$ and $K_4^{(n+192)}$ after 192 protocol runs. She can calculate $\underline{[K_2^{(n+2)}]_{96}}$:

$$[K_2^{(n+2)}]_{96} = [K_2^{(n+1)}]_{96} \oplus [n_2^{(n+1)}]_{96} \oplus [K_4^{(n+1)}]_{96} \oplus [ID]_{96} =$$

$$= ([K_2^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} \oplus [K_4^{(n)}]_{96} \oplus [ID]_{96}) \oplus [n_2^{(n+1)}]_{96} \oplus$$

$$\oplus([K_4^{(n)}]_{96} \oplus [n_1^{(n)}]_{96} \oplus [K_2^{(n)}]_{96} \oplus [ID]_{96}) \oplus [ID]_{96} =$$

$$= [n_1^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} \oplus [n_2^{(n+1)}]_{96} \oplus [ID]_{96}.$$

Clearly, she can calculate $\underline{[K_4^{(n+2)}]_{96}}$ on a similar way, and can get the following:

$$[K_4^{(n+2)}]_{96} = [n_1^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} \oplus [n_1^{(n+1)}]_{96} \oplus [ID]_{96}.$$

If the attacker has eavesdropped more rounds of the protocol (thus she has obtained random numbers $\underline{n_1^{(n+2)}}$, $\underline{n_2^{(n+2)}}$, $\underline{n_1^{(n+3)}}$, $\underline{n_2^{(n+3)}}$, etc. by the way described in 4.1), then she can calculate $\underline{[K_2^{(n+3)}]_{96}}$, $\underline{[K_4^{(n+3)}]_{96}}$, $\underline{[K_2^{(n+4)}]_{96}}$, $\underline{[K_4^{(n+4)}]_{96}}$, etc. by the help of the updating formulas. With this knowledge she can handle the addition modulo 2^{96} for the 95th bits in the updating formulas for $K_2^{(n+3)}$, $K_4^{(n+3)}$, $K_2^{(n+4)}$, $K_4^{(n+4)}$, etc. This means that she can repeat a similar method like the above for finding out $[K_2^{(n+4)}]_{95}$ and $[K_4^{(n+4)}]_{95}$. If the attacker continues this procedure with the 94th, 93rd, etc. bits, then she can learn one more bit of K_2 and K_4 if she has eavesdropped two more rounds. It follows that after 192 runs of the protocol, she will know $K_2^{(n+192)}$ and $K_4^{(n+192)}$.

4.2.2 Second Way

The attacker can use the above described method for learning K_2 and K_4, but she can accelerate it if she takes into account the information contained in B and D. In the previous part we have shown how an attacker can learn a new bit of K_2 and K_4 after two protocol runs. To show how one can improve on that procedure we just give an example. Let us suppose that one has calculated $[K_2^{(n+2)}]_{96}$ and $[K_4^{(n+2)}]_{96}$. If she is lucky then she knows $[K_2^{(n+1)}]_{96}$ or $[K_4^{(n+1)}]_{96}$ with the help of $B^{(n+1)}$ or $D^{(n+1)}$, as described at the beginning of 4.2. In this fortunate case, she can obtain the other one with the help of the updating formulas. For instance, if she knows $[K_2^{(n+1)}]_{96}$, then she can calculate $[K_4^{(n+1)}]_{96}$ from the updating formula of $K_4^{(n+1)}$. If she is more lucky, then she also knows $[K_2^{(n)}]_{96}$ or $[K_4^{(n)}]_{96}$ with the help of $B^{(n)}$ or $D^{(n)}$. In this case, she also can calculate the other one on a similar way to the above mentioned case. We do not go into details, but it is easily shown that these fortunate cases have the following probabilities: the attacker has no gain from this improvement with probability $\frac{1}{2}$, gains one round with probability $\frac{1}{4}$ and gains 2 rounds with probability $\frac{1}{4}$. This means that the expected value of the possible gain is $\frac{3}{4}$ rounds. It means that the expected value of the needed rounds for fully break the protocol is decreasing from 192 to 120 (=192-$\frac{3}{4}\cdot$96).

4.2.3 Third Way

The attacker has another possible way of determining K_2 and K_4. In this attack, she needs to divide the 96-bit long vectors into twelve 8-bit long blocks and first to take into account only the 8 least significant bits. All she needs to do is testing all the 2^{16} possible $K_2^{(n)} - K_4^{(n)}$ pairs. After she has fixed such a pair then she can generate $K_2^{(n+1)}$, $K_4^{(n+1)}$, $K_2^{(n+2)}$, $K_4^{(n+2)}$, $K_2^{(n+3)}$, $K_4^{(n+3)}$, etc. with the help of updating formulas and random numbers obtained by the above described method. Of course there are restrictions hidden in message B and D. Namely, as mentioned previously, about one fourth of K_2 and K_4 is given in each round. Another observation is that one can get only one pair from different initial $K_2 - K_4$ pairs after some updating. For example, if the last 7 bits of K_2 and K_4 are the same, and only the first bits differ then after updating one gets the same pairs from the previously different ones. One can observe that after a few rounds from the 2^{16} originally possible $K_2^{(n)}$ - $K_4^{(n)}$ pairs there will be only a few left. Table 1 shows that how many different pairs of $K_2^{(i+1)}$ - $K_4^{(i+1)}$ are possible in round $i+1$, after an attacker has eavesdropped i rounds of the protocol (we tested with a MATLAB-implementation 1000 times and counted the different pairs). The table shows that after 6 eavesdropped rounds of the protocol there is only one possible $K_2^{(n+6)}$ - $K_4^{(n+6)}$ pair with probability approx 0.9. The expected value of the needed rounds for having a unique K_2 - K_4 pair is less than 4.5 according to our simulations.

# different pairs	1	2	3	4	5	6	>6
after 3 rounds	90	144	103	123	44	91	405
after 4 rounds	426	308	101	59	29	31	46
after 5 rounds	714	220	39	13	7	4	3
after 6 rounds	904	88	6	1	1	0	0

TABLE 1 - NUMBERS OF POSSIBLE K_2 - K_4 PAIRS ACCORDING TO THE TEST-SIMULATION

If the attacker knows the least significant 8 bits of $K_2^{(i)}$ and $K_4^{(i)}$ then she can handle the modulo 2^{96} addition for the 88th bits in the updating formulas of $K_2^{(i+1)}$ and $K_4^{(i+1)}$. Hence, she can repeat the whole procedure for bits produce indices between 81 and 88, and so on. There are twelve 8-bit blocks, so the attacker is able to get K_2 and K_4 after eavesdropped about 54 rounds of the protocol (=12 $\cdot$ 4.5).

V. ATTACKS

In this section we summarize the possible attacks based on our method.

1. Tracking After eavesdropping two consecutive rounds of the protocol, the attacker learns the unique identification number ID of the tag, as described in Section 4.1.3. Thus the tags can be tracked by unauthorized parties.

Remember that M^{2}AP is a mutual authentication protocol. Next we point out that one can attack the authentication of both directions.

2. Impersonating the Reader The attacker can successfully impersonate the reader and authenticate itself to a legitimate tag in the ith round without knowing $K_2^{(i)}$ and $K_4^{(i)}$. It works, because in this case she can choose random numbers $n_1^{(i)}$ and $n_2^{(i)}$ appropriately. Only those bits of $K_2^{(i)}$ are relevant in message $B^{(i)}$, which have an index from the following set:

$$\mathcal{B}^{(i)} := \{1 \leq k \leq 96 \mid ([IDS^{(i)}]_k = 1) \wedge ([n_1^{(i)}]_k = 0)\}.$$

For the sake of simplicity we introduce the following sets:

$$\mathcal{I}_0^{(i)} := \{1 \leq k \leq 96 \mid [IDS^{(i)}]_k = 0\},$$

$$\mathcal{I}_1^{(i)} := \{1 \leq k \leq 96 \mid [IDS^{(i)}]_k = 1\},$$

$$\mathcal{K}_2^{(i)} := \{1 \leq k \leq 96 \mid \text{the attacker knows } [K_2^{(i)}]_k\}.$$

When generating $n_1^{(i)}$, the attacker needs to set $[n_1^{(i)}]_k$ to a set bit for all $k \in \mathcal{I}_1^{(i)} \backslash \mathcal{K}_2^{(i)}$. Other bits of $n_1^{(i)}$ and the whole $n_2^{(i)}$ can be chosen arbitrarily.

3. Impersonating the Tag The attacker can impersonate the tag and authenticate herself to a legitimate reader in the ith round without knowing the whole $K_2^{(i)}$ and $K_4^{(i)}$. She does not need to know anything about $K_2^{(i)}$. Furthermore, only those bits of $K_4^{(i)}$ are relevant in message $D^{(i)}$, which has an index from the following set:

$$\mathcal{D}^{(i)} := \{1 \leq k \leq 96 \mid ([IDS^{(i)}]_k = 0) \wedge ([n_2^{(i)}]_k = 1)\}.$$

We also introduce the following set:

$$\mathcal{K}_4^{(i)} := \{1 \leq k \leq 96 \mid \text{the attacker knows } [K_4^{(i)}]_k\}.$$

With this notation, the attacker can be successful in the ith round if and only if $\mathcal{D}^{(i)} \subseteq \mathcal{K}_4^{(i)}$. It offers a relatively good chance, if the attacker knows more than 80 bits of $K_4^{(i)}$.

At last we present an active attack.

4. Active Attack The attacker has a good chance to obtain the bits of K_4, if she applies active attack. She needs to choose the bits of n_2 appropriately. Namely, $[n_2^{(i)}]_k$ should be a set bit for all $k \in \mathcal{I}_0^{(i)}$. With this manipulation she can obtain about half of $K_4^{(i)}$ in one round. It results that she can reveal the secrets faster than in the case when she is fully passive.

VI. CONCLUSIONS

We have given a constructive proof that M^{2}AP is weak and can be broken. From a broader perspective, our paper once again demonstrates that various "proofs of security" based on statistical pseudo-random properties of the messages available for eavesdropping are meaningless. Such properties are neither sufficient nor necessary for the security of a communication system in any meaningful sense.

When demonstrating the (computational) security of a system, researchers should show that the ability to breach it implies the ability to solve a computational problem that is believed to be unfeasible, which is of course a condition upon which the security assumption depends. In some cases, it is possible to prove security unconditionally by demonstrating that the mutual information between the observable and the secret parameters equals zero.

REFERENCES

[1] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan Estevez-Tapiador, and Arturo Ribagorda. M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *International Conference on Ubiquitous Intelligence and Computing – UIC'06*, volume 4159 of *Lecture Notes in Computer Science*, pages 912–923. Springer-Verlag, September 2006.

[2] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan Estevez-Tapiador, and Arturo Ribagorda. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. Printed handout of Workshop on RFID Security – RFIDSec 06, July 2006.

[3] Mihály Bárász, Balázs Boros, Péter Ligeti, Krisztina Lója, and Dániel A. Nagy. Breaking LMAP. Printed handout of Workshop on RFID Security – RFIDSec 07, July 2007.

[4] Tieyan Li and Guilin Wang. Security analysis of two ultra-lightweight RFID authentication protocols. In *IFIP SEC 2007*, Sandton, Gauteng, South Africa, May 2007. IFIP.

[5] ELTECRYPT Research Group homepage. http://www.cs.elte.hu/eltecrypt/.