Breaking LMAP [⋆]

Mihály Bárász[1], Balázs Boros[1], Péter Ligeti[1,2], Krisztina Lója[1,3], Dániel A. Nagy[1]

[1] ELTECRYPT Research Group
Department of Computer Science, Eötvös University
1117 Budapest, Pázmány Péter sétány 1/c, Hungary
[2] Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences
1053 Budapest, Reáltanoda u. 13-15, Hungary
[3] Department of Telecommunications and Telematics,
Budapest University of Technology and Economy
1117 Budapest, Magyar Tudósok krt. 2, Hungary

Abstract. In this paper, we present a passive attack resulting the complete break of LMAP (Lightweight 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. Assuming that the attacker can eavesdrop successive authentication rounds of the same RFID tag, after a small number of rounds (the expected number is about 10) the attacker learns the identification number of the tag and every common secret shared by the tag and the reader. This means that in the subsequent rounds the attacker can successfully impersonate the targeted tag.

Keywords: RFID, Tag, Reader, Mutual Authentication, LMAP, Passive Attack

1 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 that might want to impersonate either tags or readers without being authorized to do so.

In the particular case of LMAP, 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.

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

LMAP 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 LMAP certainly falls short of achieving it.

Li and Wang [2] have pointed out some weaknesses of LMAP. 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.

In this paper, we show a fully passive attack against the protocol, which is able to get all the secret information stored in the tag only by eavesdropping a few consecutive rounds of the protocol. The rest of this paper is organized as follows. In Section 2, we introduce the LMAP protocol of Peris-Lopez et al [1]. In Section 3, we point out the main weaknesses of the protocol. In Section 4 we present our passive attack step by step. In Section 5, we give some supplements to the previous section. In Section 6, we briefly introduce the active attack against LMAP based on [2], and offer remarks on it. Finally, in Section 7, we summarize our results.

2 The Protocol

We give a brief introduction to LMAP. 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 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 ($\vee$) 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:

1. **Tag Identification**

$$\boxed{\text{READER}} \rightarrow \textit{hello} \rightarrow \boxed{\text{TAG}}$$

$$\boxed{\text{READER}} \leftarrow IDS^{(n)} \leftarrow \boxed{\text{TAG}}$$

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 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**

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

$$\boxed{\text{READER}} \rightarrow B^{(n)} := (IDS^{(n)} \vee K_2^{(n)}) + n_1^{(n)} \rightarrow \boxed{\text{TAG}}$$

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

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**

$$\boxed{\text{READER}} \leftarrow D^{(n)} := (IDS^{(n)} + ID) \oplus n_1^{(n)} \oplus n_2^{(n)} \leftarrow \boxed{\text{TAG}}$$

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

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

$$IDS^{(n+1)} := (IDS^{(n)} + (n_2^{(n)} \oplus K_4^{(n)})) \oplus ID$$

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

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

$$K_3^{(n+1)} := (K_3^{(n)} \oplus n_1^{(n)}) + (K_1^{(n)} \oplus ID)$$

$$K_4^{(n+1)} := (K_4^{(n)} \oplus n_1^{(n)}) + (K_2^{(n)} \oplus ID)$$

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

3 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 LMAP 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 subsection 4.1 we do not take difference between $\oplus$ and $+$.

The bitwise OR operation in message B is another weak point of the protocol. From message B, one can easily gain information about the random number n_1, especially with the help of the set bits of IDS.

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

4 The steps of breaking the LMAP

Let us denote the kth bit of M in round n by $[M^{(n)}]_k$ for $M \in \{A, B, C, D, K_1, K_2, K_3, K_4, IDS, n_1, n_2\}$. 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 $A^{(n)}, B^{(n)}, C^{(n)}, D^{(n)}$ 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. $\underline{[n_1^{(n)}]_k}$).

The attacker will need to eavesdrop a few consecutive rounds of authentication of the same tag. First, we summarize the steps of our attack, afterwards we explain the steps precisely.

First, the attacker calculates the least significant bits of every key and secret. It is easy because the XOR operator and the addition mod 2^{96} are the same with respect to the least significant bits. The only necessary thing is that the least significant bit of IDS could be a set bit. If this is not the case, the attacker simply waits for another round of authentication.

Once the attacker receives an IDS ending with a set bit, after two more rounds, the least significant bits of all the unknowns in all eavesdropped rounds get revealed (to the attacker). The next step is to calculate the bits immediately before the least significant ones with the knowledge of the latter. At this point, the attacker will be able to set up the equations using just XOR, without addition mod 2^{96}.

Thus, step by step the attacker will learn all the bits from the least significant ones to the most significant ones. The attacker needs to eav esdrop r rounds such that for every k: $[IDS^{(n)}]_k \vee [IDS^{(n+1)}]_k \vee [IDS^{(n+2)}]_k \vee \cdots \vee [IDS^{(n+r-1)}]_k = 1$ and two more rounds for calculating the bits where the IDS in rounds n, $n+1$, $\ldots$, $n+r-2$ are 0 and become 1 in round $n+r-1$, see more precisely in Section 5.

4.1 The Least Significant Bits

Let us assume that $[IDS^{(n)}]_{96} = 1$. This implies that $[IDS^{(n)}]_{96} \vee [K_2^{(n)}]_{96} = 1$. From message B, the attacker can compute $\underline{[n_1^{(n)}]_{96}}$:

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

From message A she can get $\underline{[K_1^{(n)}]_{96}}$:

$$[K_1^{(n)}]_{96} = [A^{(n)}]_{96} \oplus [n_1^{(n)}]_{96} \oplus 1.$$

From message C and D she can get $\underline{[K_3^{(n)}]_{96} \oplus [n_2^{(n)}]_{96}}$ and $\underline{[ID]_{96} \oplus [n_2^{(n)}]_{96}}$, respectively:

$$[K_3^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} = [C^{(n)}]_{96} \oplus 1,$$

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

From the above two equalities, she can get $\underline{[ID]_{96} \oplus [K_3^{(n)}]_{96}}$:

$$[ID]_{96} \oplus [K_3^{(n)}]_{96} = [C^{(n)}]_{96} \oplus [D^{(n)}]_{96} \oplus [n_1^{(n)}]_{96}.$$

We assume that the attacker is able to eavesdrop more rounds of authentication, one after the other, so after round $n+1$, the attacker also knows the least significant bits of $IDS^{(n+1)}$, $A^{(n+1)}$, $B^{(n+1)}$, $C^{(n+1)}$ and $D^{(n+1)}$ as well. From the definition of updating the index-pseudonym, she can obtain $\underline{[K_4^{(n)}]_{96}}$:

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

Considering the equation of message $C^{(n+1)}$ and the updating formula of K_3, she can obtain $\underline{[n_2^{(n+1)}]_{96}}$:

$$[n_2^{(n+1)}]_{96} = [K_3^{(n+1)}]_{96} \oplus [IDS^{(n+1)}]_{96} \oplus [C^{(n+1)}]_{96} =$$

$$= \big([n_1^{(n)}]_{96} \oplus [K_1^{(n)}]_{96} \oplus ([K_3^{(n)}]_{96} \oplus [ID]_{96})\big) \oplus [IDS^{(n+1)}]_{96} \oplus [C^{(n+1)}]_{96}.$$

Considering the updating formula of K_4 and IDS, she can obtain $\underline{[K_2^{(n)}]_{96}}$:

$$[K_2^{(n)}]_{96} = [K_4^{(n)}]_{96} \oplus [n_1^{(n)}]_{96} \oplus [K_4^{(n+1)}]_{96} \oplus [ID]_{96} =$$

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

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

Now she can calculate $\underline{[K_2^{(n+1)}]_{96}}$ by the respective updating formula:

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

From message B she can obtain $\underline{[n_1^{(n+1)}]_{96}}$:

$$[n_1^{(n+1)}]_{96} = [B^{(n+1)}]_{96} \oplus ([IDS^{(n+1)}]_{96} \vee [K_2^{(n+1)}]_{96}.$$

Now, from message D she can obtain $\underline{[ID]_{96}}$:

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

From the message D she can get $\underline{[n_2^{(n)}]_{96}}$:

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

And finally, from message C she can get $\underline{[K_3^{(n)}]_{96}}$:

$$[K_3^{(n)}]_{96} = [C^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} \oplus 1.$$

In this subsection we have pointed out the following: if the eavesdropper knows the least significant bits of $IDS^{(n)}$, $A^{(n)}$, $B^{(n)}$, $C^{(n)}$, $D^{(n)}$, $IDS^{(n+1)}$, $A^{(n+1)}$, $B^{(n+1)}$, $C^{(n+1)}$, $D^{(n+1)}$ and $IDS^{(n+2)}$ and in addition $[IDS^{(n)}]_{96} = 1$, then she can obtain the least significant bits of the secrets in the nth, $(n+1)$th and $(n+2)$th round of the protocol by using the above equations and the updating formulas. In addition, if she eavesdrops the $(n+3)$th protocol run, then she will be able to determine $n_1^{(n+3)}$, $n_2^{(n+3)}$ and also the keys in the $(n+4)$th round of the protocol, since she knows everything that the tag knows, and so on. If $[IDS^{(n)}]_{96} = 0$, then the attacker needs to wait for another round of the protocol. She can use the above described method, if $[IDS^{(n+s)}]_{96} = 1$ for some $s \in \mathbf{N}$. The only thing she needs to do is write $n+s$ everywhere instead of n.

The following remark is very simple, but important: if $[IDS^{(n+1)}]_{96} = 1$, then the attacker can obtain the least significant bits of every secret also in round n. In this case she can obtain the least significant bits of every secret in round $n+1$. We do not go into the details, but she can use the following equations:

$$\underline{[K_3^{(n)}]_{96}} = [K_3^{(n+1)}]_{96} \oplus ([n_1^{(n)}]_{96} \oplus [K_1^{(n)}]_{96}) \oplus [ID]_{96},$$

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

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

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

$$\underline{[n_1^{(n)}]_{96}} = [D^{(n)}]_{96} \oplus [IDS^{(n)}]_{96} \oplus [n_2^{(n)}]_{96} \oplus [ID]_{96},$$

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

Clearly, the above remark also holds for $n+s-1$ and $n+s$ instead of n and $n+1$.

4.2 The bits immediately before the least significant ones

By using the method described in the previous subsection, the attacker will determine the 95th bits.

The attacker can set up all of the equations for the 95th bits by using the 96th bits. The main point is that the attacker can handle the addition modulo 2^{96} for the kth bit if she knows the jth bits of the addends for every $k < j \leq 96$. For example, if $[IDS^{(n)}]_{96} \wedge [ID]_{96} = 1$ then the equation of $[D^{(n)}]_{95}$ gets the following form:

$$[D^{(n)}]_{95} = [IDS^{(n)}]_{95} \oplus [ID]_{95} \oplus [n_1^{(n)}]_{95} \oplus [n_2^{(n)}]_{95} \oplus 1.$$

Let us assume that $[IDS^{(n)}]_{95} = 1$. In this case, the attacker can obtain the bit $[n_1^{(n)}]_{95}$ from message B, and after that she can calculate everything with the same method as in the previous subsection, and she does not need to eavesdrop other runs of the protocol, the so far eavesdropped messages are sufficient.

Now let us assume that $[IDS^{(n)}]_{95} = 0$ and $[IDS^{(n+1)}]_{95} = 1$. In this case, the attacker needs to eavesdrop one more run of the protocol. Precisely, she has to eavesdrop the nth, $(n+1)$th and $(n+2)$th protocol run and $IDS^{(n+3)}$. After that she is again able to use the same method as in the previous subsection, and can get the values of the keys in the $(n+1)$th, $(n+2)$th and $(n+3)$th rounds. In addition, she can get the values of the keys in round n. For example, she can obtain $\underline{[K_3^{(n)}]_{95}}$ with the help of $[K_3^{(n+1)}]_{95}$, $[K_1^{(n)}]_{95} \oplus [n_1^{(n)}]_{95}$ and $[ID]_{95}$. After that she can obtain $\underline{[n_1^{(n)}]_{95}}$ from message $[C^{(n)}]_{95}$, $[IDS^{(n)}]_{95}$ and $[K_3^{(n)}]_{95}$, and so on.

4.3 More significant bits

Clearly, the attacker can derive all the secrets by the same method as she obtained the least significant bits and the bits immediately before the least significant bits. The only thing she needs is a set bit in IDS. In this subsection we formalize this sentence precisely and in Section 5, we add more explanation.

If p and q are fixed integers then, for the sake of simplicity, we use the following notation:

$$[p, q] := \{l \in \mathbf{Z} | p \leq l \leq q\}.$$

Let us denote the following random variable, which is non-negative and has integer values, by r:

$$r := inf\{r^0 \in \mathbf{N} | \forall\ k \in [1, 96]\ \exists\ i \in [0, r^0 - 1] : [IDS^{(n+i)}]_k = 1\}.$$

Taking into account all the findings of Section 4, the attacker can consider that after eavesdropping $r + 1$ consecutive runs of the protocol (note that r is a random variable) and $IDS^{(n+r+1)}$ (the next IDS), she can obtain all the originally unknown parameters. Namely, all elements of the following set:

$$\left\{M^{(n+i)} \in \{0,1\}^{96} | M \in \{K_1, K_2, K_3, K_4, n_1, n_2, IDS\}, i \in [0, r+1]\right\} \bigcup \{ID\}.$$

This means that the attacker knows all the information contained by the tag. Thus, she can also update IDS and the secret keys. With this knowledge, she can fully impersonate the tag in the $(n + r + 2)$th round of the protocol. The tag no longer has any secret unknown to the attacker. In Section 5, we provide more details about the random variable r.

5 Supplement

In this section, we calculate the distribution and the expected number of random variable r introduced in the previous section.

We can assume the following (supposing that the nth run of the protocol is not the first one):

$$\mathbf{P}(\{[IDS^{(n)}]_k = 0\}) = \mathbf{P}(\{[IDS^{(n)}]_k = 1\}) = 1/2 \quad (\forall\, k \in [1, 96]).$$

The above equation holds, because the updating of the index-pseudonym contains n_2 and we can make the following assumption: n_2 is uniformly chosen from $\{0, 1\}^{96}$. Of course the same holds for $n+i$ instead of n, where $i \in \mathbf{N}$. Under these assumptions, we can evaluate the expected value of r. For this purpose, we introduce random variables: r_j ($j \in [1, 96]$), such that

$$r_j := inf\{r_j^0 \in \mathbf{N} | [IDS^{(n+r_j^0-1)}]_j = 1\}.$$

With this definition, we get the following:

$$r = max\{r_j | j \in [1, 96]\}.$$

It is easy to show that $\mathbf{P}(\{r_j \leq k\}) = 1 - \frac{1}{2^k}$ for any $k \in \mathbf{N}$ and for any $j \in [1, 96]$, because r_j's have geometric distribution with parameter $\frac{1}{2}$. Since r_j's are iid, we obtain the following equations for any integer $k \in \mathbf{N}$:

$$\mathbf{P}(\{r \leq k\}) = \mathbf{P}(\{r_j \leq k, \forall\, j \ \in [1, 96]\}) = \prod_{j=1}^{96} \mathbf{P}(\{r_j \leq k\}) = \Big(1 - \frac{1}{2^k}\Big)^{96}.$$

From the above equations, we can compute the probability $\mathbf{P}(\{r = k\})$ for $k \in \mathbf{N}$:

$$\mathbf{P}(\{r = k\}) = \mathbf{P}(\{r \leq k\}) - \mathbf{P}(\{r \leq k - 1\}) = \Big(1 - \frac{1}{2^k}\Big)^{96} - \Big(1 - \frac{1}{2^{k-1}}\Big)^{96}.$$

Finally, we can obtain the expected value of r, calculating the following sum:

$$\mathbf{E}(r) = \sum_{k=1}^{\infty} k \Bigg(\Big(1 - \frac{1}{2^k}\Big)^{96} - \Big(1 - \frac{1}{2^{k-1}}\Big)^{96}\Bigg).$$

We can calculate the sum numerically, it will be about 7.93.

k	1	2	3	4	5	6	7
P({r=k})	$< 10^{-28}$	$< 10^{-11}$	$< 10^{-5}$	$< 10^{-2}$	0.05	0.17	0.25
k	8	9	10	11	12	13	14
P({r=k})	0.22	0.14	0.08	0.04	0.02	0.01	$< 10^{-2}$

Table 1. Distribution of r

The distribution of the random variable r is presented in Table 1.

6 Active Attack Against LMAP by Li and Wang

Li and Wang have presented an active attack against LMAP. In this subsection we mention some of the main points of that paper. For more details, please refer to [2].

The authors have introduced the following two attacks: *De-synchronization Attack* and *Full Disclosure Attack*. In this paper we do not deal with the first one, only with the second one.

First, the authors remark that the tag does not know if D is indeed received or verified by a legitimate reader. This can cause difference between the storage of the tag and the reader, so they can fall out of synchronization. To avoid this problem, they suppose there is a completion message being sent to each other to indicate a successful completion of the protocol.

The authors also take the following supposition: the tag has no memory for status information (therefore it is considered stateless), but a legitimate reader is stateful (as to remember all status information regarding the protocol with a specific tag). This means that one can repeatedly run the incomplete protocol many times at the tag side. This assumptions is reasonable as the tag has to answer any request by legitimate or illegitimate readers, and the protocol is not complete if the reader did not send completion message to the reader.

After taking these assumptions, we mention some details about the Full Disclosure Attack. Under the assumption that the tag is stateless, they can interrogate the tag many times. It means that they can send the tag A, B, C triplets many times. They send exactly 96 different such triplets, and according to whether a proper D or an error message is received, the attacker concludes one bit of n_1. So after this step the attacker will know n_1. The authors claim that from A, B, IDS and n_1, the attacker can calculate K_1 and K_2. However, one cannot get K_2, only "half of K_2".

The next step is obtaining ID. To this aim, they interact once again with the reader and the tag also. After these interactions, one need to solve the following equation:

$$x \oplus a = x \oplus b + c \mod 2^{96},$$

where $a, b, c \in \{0,1\}^{96}$ are known vectors and $x \in \{0,1\}^{96}$ is unknown. If the attacker get x, then she can easily calculate ID. But it is important to remark that there is no guarantee for the uniqueness of solution x. The authors suggest a trivial way to solve this equation, but that requires too much computing time. They also mention the paper of Lipmaa and Moriai [3], where a method is given for solving the equation in complexity $\mathcal{O}(m)$, where m is the length of the vectors, so in our case $m = 96$. The authors claim that if there are more than one possible ID, then one can repeat the whole attack later and can get other possible values for ID. Of course, the real ID satisfies all of the occurrent equations. Once the value of ID is fixed, one can easily obtain K_3 and K_4. The latter from the next IDS. But that is not clear how they can derive the "missing half of K_2".

7 Conclusions

We have given a constructive proof that LMAP is weak and can be broken. First, we have shown that assuming that the least significant bit of $IDS^{(n)}$ is a set bit, the attacker is able to calculate the least significant bits of every key and secret of round n after eavesdropping the communication between the tag and the reader in rounds n, $n+1$ and $n+2$. We have shown that assuming $[IDS^{(n)}]_k = 1$ and bits $k+1$, $k+2$, $\dots$, 96 of every key and secret are known, the attacker can obtain the kth bits of every key and secret if the communication between the tag and the reader in rounds n, $n+1$ and $n+2$ are eavesdropped.

It obviously follows that knowing the kth bits in round n and every bit between $k+1$ and 96 in rounds n, $n+1$, $\dots$, $n+s$, one can calculate the kth bits in $n+1$, $n+2$, $\dots$, $n+s$. We have shown that the attacker can also calculate the bits between k and 96 in rounds n, $n+1$, $\dots$, $n+s-1$ with the knowledge of the bits between k and 96 in round $n+s$.

We also have shown that the expected number of rounds need to eavesdrop is about 10.

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 infeasible, 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.

Our passive attack is robust 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.

References

1. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, A. Ribagorda: LMAP: A Real Lightwight Mutual Authentication Protocol for Low-cost RFID tags. Proceedings of RFIDSec06 Workshop on RFID Security, 12-14 July, Graz, Austria, (2006)
2. T. Li, G. Wang: Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols. IFIP SEC 2007, 14-16 May, Sandton, Gauteng, South Africa, (2007)
3. H. Lipmaa, S. Moriai: Efficient Algorithms for Computing Differential Properties of Addition. Proceedings of FSE '01, **LNCS 2355** (2001) 336–350.