1-1-2007

Security analysis of two ultra-lightweight RFID authentication protocols

Tieyan Li
Institute for Infocomm Research

Guilin Wang
Institute for Infocomm Research, guilin@uow.edu.au

Recommended Citation

Security analysis of two ultra-lightweight RFID authentication protocols

Abstract

In this paper, we analyze the security vulnerabilities of two ultra-lightweight RFID mutual authentication protocols: LMAP and M2AP, which are recently proposed by Peris-Lopez et al.We identify two e®ective attacks, namely De-synchronization attack and Full-disclosure attack, against their protocols. The former attack can break the synchro- nization between the RFID reader and the tag in a single protocol run so that they can not authenticate each other in any following protocol runs. The latter attack can disclose all the secret information stored on a tag by interrogating the tag multiple times. Thus it compromises the tag completely. Moreover, we point out the potential countermeasures to improve the security of above protocols

Keywords

rfid, lightweight, protocols, ultra, authentication, two, analysis, security

Disciplines

Engineering | Science and Technology Studies

Publication Details

Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols

Tieyan Li and Guilin Wang

Systems and Security Department
Institute for Infocomm Research (I^2R)
21 Heng Mui Keng Terrace, Singapore 119613
{litieyan, glwang}@i2r.a-star.edu.sg

Abstract. In this paper, we analyze the security vulnerabilities of two ultra-lightweight RFID mutual authentication protocols: LMAP and M^2AP, which are recently proposed by Peris-Lopez *et al.* We identify two effective attacks, namely *De-synchronization attack* and *Full-disclosure attack*, against their protocols. The former attack can break the synchronization between the RFID reader and the tag in a single protocol run so that they can not authenticate each other in any following protocol runs. The latter attack can disclose all the secret information stored on a tag by interrogating the tag multiple times. Thus it compromises the tag completely. Moreover, we point out the potential countermeasures to improve the security of above protocols.

1 Introduction

Radio Frequency Identification (RFID) systems have been aggressively deployed in a variety of applications, but their further pervasive usage is mainly limited by a number of security and privacy concerns. Since RFID tags are generally low cost with extremely limited resources, traditional security primitives can not be incorporated well. But when they are deployed in pervasive environment, where threats are not uncommon, security and privacy issues must be addressed before their massive deployment (refer to Section 2).

In this paper, we analyze the security of two ultra-lightweight RFID mutual authentication protocols, *w.r.t.,* LMAP [14] and M^2AP [15], which are recently proposed by Peris-Lopez *et al.* Different from the majority of existing solutions [17, 13, 11, 12, 1] of using classic cryptographic primitives, those two protocols are *ultra-lightweight*, since they use only simple bitwise operations to achieve mutual authentication between the RFID reader and the tags. Consequently, only about 300 gates are required to implement such an RFID tag. The protocols are very practical to be implemented on low-cost tags (with price $0.05 - 0.1$ US dollar), where less than $1K$ (out of totally $5K$) gates are allowed for security operations. Moreover, both LMAP and M^2AP protocols are claimed to be secure in sense of "Man-in-the-middle attack prevention" and "forgery resistance". However, we identify some vulnerabilities in those two protocols. Specifically,

we first show that the protocols suffer from *De-synchronization attack*. The attack is very effective by only eavesdropping a single protocol run and then can destroy the "synchronization" between the database [1] and the tag. Thus, the tag cannot be further authenticated by the database. Then we present a more serious attack - *Full-disclosure attack*. By interacting with the reader ($O(1)$ times) and the tag ($O(m)$ times), this attack enables an attacker to compromise the ID of the tag, as well as all other secret information stored on a tag. Thus, all security properties claimed by above protocols are destroyed. Finally, to defend against the above attacks, we propose several potential countermeasures. One of them addresses the stateless property of the original protocols, which could be enhanced by adding status information into the protocols. As a result, additional ($\sim 40\%$) memory space is needed to implement such a tag.

The rest of this paper is organized as follows. Section 2 generally reviews the related work on RFID security and privacy issues. Then, we review LMAP and M^{2}AP in Section 3 and analyze their vulnerabilities in Section 4. Section 5 points out several countermeasures. At last, we conclude the paper.

2 Security and privacy issues in RFID systems

2.1 Threat Model

The proliferation of RFID tags implies that pervasive RFID technology might bring unintended risks. Actually, more than hundreds of research papers were published on addressing RFID security and privacy problems (please refer to [2, 9, 5] for literature survey). Some of the earlier research works draw assumptions on practical limitations of RFID deployments: *i.e.,* firstly, considering the tag-to-reader channel is private, since the backscatter channel from the tag to the reader has a relatively shorter range (*e.g.,* several centimeters) than that of the forward channel. Thus, an attacker, not within the range, cannot get reply from the tag. And secondly, it is not easy for an attacker to hide himself between a legitimate reader and a tag in an active session. Thus, there is no man-in-the-middle between the tag and the reader. Also thirdly, it is not easy to intercept a message and modify the message over the air in real time, because of shared wireless bearing medium. While these assumptions do make sense in many practical RFID deployments to provide reasonable security, they are not strict and sufficient on addressing the threats encountered in strict deployment environments. Most recently, the research works [8, 3] assume the most reasonable (strongest) threat model that an active man-in-the-middle can eavesdrop, intercept or modify messages in real time on both forward and backward channels. And the protocols in [14] and [15] assume the man-in-the-middle attack, which rationalizes our work.

---

[1] As in [14] and [15], the protocols make no difference on the database and the reader. Thereafter, we use either the reader or the database to indicating the counterpart of a tag in the authentication protocols.

2.2 RFID Authentication Protocols

Authentication normally involves the use of secret data. Note that not all RFID tags are able to be authenticated since their inabilities of storing the secret data, *e.g.*, EPC class I tags. In the literature, one widely adopted assumption is using hash function within a tag. Weis *et al.* propose a randomized "hash lock" based mutual authentication protocol in [17]. Ohkubo *et al.* proposed a hash chain model embedding two hash functions in a tag [13]. Some solutions assume Pseudo-Random Function (PRF) in a tag. Molnar and Wagner use a tree scheme for authentication [11]. They further propose a scalable pseudonym protocol for ownership transfer [12]. Other assumptions includes using symmetric cipher, like [1], in which Feldhofer *et al.* proposed a simple two way challenge-response mutual authentication protocol with AES encryption algorithm. The other work [6] even assume public key cryptographic primitive, in which tags update their IDs with re-encryption scheme.

   To reduce the gate numbers in RFID tags, some approaches have been proposed without assumptions on classic cryptographic primitives. In [18], Weis introduced the concept of human computer authentication protocol due to Hopper and Blum, adaptable to low-cost RFID tags. Further on, Weis and Juels proposed a lightweight symmetric-key authentication protocol named HB$^+$ [8]. The security of both the HB and the HB$^+$ protocols is based on the Learning Parity with Noise (LPN) Problem, whose hardness over random instances still remains as an open question. In [16], the authors proposed a set of extremely-lightweight challenge-response authentication protocols that are suitable for authenticating tags, but their protocols can be broken by a powerful adversary [4]. In [7], Juels proposed a solution based on pseudonyms without using hash functions at all. The RFID tags store a short list of random identifiers or pseudonyms (known by authorized verifiers). When tag is queried, it emits the next pseudonym in the list. However, the list of pseudonyms should be reused or updated via an out-of-band channel after a number of authentications. Due to those reasons, Peris-Lopez et al. proposed two mutual authentication protocols for low-cost RFID tags: LMAP [14] and M^{2}AP [15], in which only simple bitwise operations are used. Their schemes are extremely lightweight and claimed to be secure against many attacks. However, we shall show the vulnerabilities of those protocols in the following sections.

3 Review of LMAP and M^{2}AP

In LMAP [14] protocol, simple operations such as: bitwise XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$), and addition mod 2^m ($+$), are used. Costly operations such as multiplications and hash evaluations are not required at all, and random number generation is only done by the reader. The scheme uses index-pseudonyms (IDSs). An index-pseudonym (96-bit length) is the index of a table (a row) where all the information about a tag is stored. Each tag is associated

with a key, which is divided in four parts of 96 bits ($K = K1||K2||K3||K4$). As the IDS and the key (K) must be updated, it needs 480 bits of rewritable memory (EEPROM) in total. A ROM memory to store the 96-bit static identification number (ID) is also required. The protocol is shown in Table 1.

Tag identification: Reader $\longrightarrow$ Tag: *hello* Tag $\longrightarrow$ Reader: $IDS_{tag(i)}^{(n)}$	*where:*
LMAP mutual authentication: Reader $\longrightarrow$ Tag: $A||B||C$ Tag $\longrightarrow$ Reader: D	$A = IDS_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} \oplus n1$ $B = (IDS_{tag(i)}^{(n)} \vee K2_{tag(i)}^{(n)}) + n1$ $C = IDS_{tag(i)}^{(n)} + K3_{tag(i)}^{(n)} + n2$ $D = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2$
M^2AP *mutual authentication:* Reader $\longrightarrow$ Tag: $A||B||C$ Tag $\longrightarrow$ Reader: $D||E$	*where: A, C same as in LMAP.* $B = (IDS_{tag(i)}^{(n)} \wedge K2_{tag(i)}^{(n)}) \vee n1$ $D = (IDS_{tag(i)}^{(n)} \vee ID_{tag(i)}) \wedge n2$ $E = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1$

Table 1. LMAP and M^2AP Protocol

The protocol has three main stages: tag identification, mutual authentication, index-pseudonym updating and key updating.

- *Tag Identification*: The reader sends a *hello* message to the tag, which will reply with its current index-pseudonym (IDS). By means of this IDS, only an authorized reader is able to access the tag's corresponding secret key ($K = K1||K2||K3||K4$), which is necessary to carry out the next authentication stage.
- *Mutual Authentication*: The reader first generates two random numbers $n1$ and $n2$. With $n1$, $n2$, and the subkeys $K1$, $K2$ and $K3$, the reader generates the submessages A, B and C, and then sends them to the tag. With the submessages A and B, the tag can authenticate the reader and obtain $n1$. Once the reader is authenticated, the tag can obtain the random number $n2$ from the submessage C, and then generates the answer message D. In this way, the tag's static identifier is transmitted securely to the read. The tag is successfully authenticated, if the reader can find a valid ID from message D. Note that the random numbers $n1$ and $n2$ are also used to update the index-pseudonym and key.
- *Index-Pseudonym and Key Updating*: After the reader and the tag authenticated each other, they carry out the index-pseudonym and key updating by the following equations.

$$IDS_{tag(i)}^{(n+1)} = (IDS_{tag(i)}^{(n)} + (n2 \oplus K4_{tag(i)}^{(n)})) \oplus ID_{tag(i)}$$
$$K1_{tag(i)}^{(n+1)} = K1_{tag(i)}^{(n)} \oplus n2 \oplus (K3_{tag(i)}^{(n)} + ID_{tag(i)})$$
$$K2_{tag(i)}^{(n+1)} = K2_{tag(i)}^{(n)} \oplus n2 \oplus (K4_{tag(i)}^{(n)} + ID_{tag(i)})$$

$$K3_{tag(i)}^{(n+1)} = (K3_{tag(i)}^{(n)} \oplus n1) + (K1_{tag(i)}^{(n)} \oplus ID_{tag(i)})$$
$$K4_{tag(i)}^{(n+1)} = (K4_{tag(i)}^{(n)} \oplus n1) + (K2_{tag(i)}^{(n)} \oplus ID_{tag(i)})$$

LMAP [14] has a sister protocol called M^2AP [15], which is a very similar lightweight RFID mutual authentication protocol. The index-pseudonym updating equation in M^2AP is changed to $IDS_{tag(i)}^{(n+1)} = (IDS_{tag(i)}^{(n)} + (n2 \oplus n1)) \oplus ID_{tag(i)}$, slightly different from LMAP. All key updating operations are the same as LMAP. The table 1 describes M^2AP, too.

The authors of [14, 15] presented some security analysis and claimed that both LMAP and M^2AP are secure against the followings: *tag anonymity, mutual authentication, man-in-the-middle attack prevention, replay attack prevention, forgery resistance.* In the next section, we identify effective attacks that can break above protocols and show the flaws with all of their claims.

4 Vulnerabilities of LMAP and M^2AP

First of all, we remark that the above protocols are not robust in the sense of cryptographic protocols, because the tag doesn't know if D is indeed received or verified by a legitimate reader. If D is not received or verified successfully, the reader will not update its storage relating to the tag, while the tag will update its storage since it has already authenticated the reader. Obviously, the storages at the tag and the reader are not synchronized. But this issue is more about an assumption problem (See more discussions on threat model in section 2.1.), not as a serious security problem. To patch it implicitly, we suppose there is a completion message being sent to each other to indicate a successful completion of the protocol. This completion message will enable the updating operations at both the reader and the tag side. All the following attacks assume that the protocols have above completion message to trigger the updating. Follow on, we will present the security problems of LMAP as well as M^2AP.

4.1 *De-synchronization Attack*

To provide privacy for an RFID tag, most RFID authentication protocols update a tag's ID after a successful protocol round. Typically, the database has to update the tag's ID accordingly so that a legitimate reader can still authenticate the tag later on. So the synchronization of secret information between the database and the tag is crucial for their following successful protocol runs. A flawed protocol, as discussed above, might leave the protocols uncomplete and cause the asynchronization at both sides. Additionally, an intended attack, like the *De-synchronization attack* introduced below, may also destroy the authentication protocols.

Attack 1: Changing message C. We now present the simplest *de-synchronization attack*: without any previous knowledge of any former protocol, a man-in-the-middle can first eavesdrop on the on-going protocol, and then change $A||B||C$ to $A||B||C'$, where $C' = C \oplus [I]_0$ and $[I]_0 = [000 \cdots 001]$ (set the first 95 most significant bits of I as 0 and the least significant bit as 1). Similarly, the attacker changes the reply D from the tag to $D' = D \oplus [I]_0$. This procedure is drawn in Table 2.

LMAP mutual authentication:	*where:*
	$n2' \leftarrow\!- n2$
Reader $\longrightarrow$ Tag: $A||B||C'$	$C' = C \oplus [I]_0$
Tag $\longrightarrow$ Reader: D'	$D = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2'$
	$D' = D \oplus [I]_0$

Table 2. *De-synchronization Attack* against LMAP

At the tag side, the attack doesn't affect the first round of interaction protocol: "tag identification". But in the second round, when the tag receives the message $A||B||C'$, it can still authenticate the reader as A and B are retained. But, the tag will get a wrong random number $n2' \leftarrow\!- n2$ (where $n2'$ depends on $n2$, but is not necessarily expressed as a function of $n2$, according to equations 1-4). The tag will accept this value and compute its reply according to $n2'$, $D = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2'$. In this simplest attack, the attacker can now provide the reader with a reply D'. If the reader accepts the value D', we say the attack is successful; otherwise, the attack is failed. Now we analyze the success rate as follows: the operation on C is actually toggling the least significant bit of C (denoted as $[C]_0$).

$$If \quad [C]_0 = 1; \Rightarrow [C']_0 = 0; \rightarrow If \quad [n2]_0 = 0, HW(n2 \oplus n2') \geq 2 \qquad (1)$$
$$\rightarrow If \quad [n2]_0 = 1, n2' = n2 \oplus [I]_0 \qquad (2)$$
$$If \quad [C]_0 = 0; \Rightarrow [C']_0 = 1; \rightarrow If \quad [n2]_0 = 0, n2' = n2 \oplus [I]_0 \qquad (3)$$
$$\rightarrow If \quad [n2]_0 = 1, HW[n2 \oplus n2'] \geq 2 \qquad (4)$$

Here, $HW(a)$ is the hamming weight of a, so $HW(a \oplus b)$ denotes the number of bit differences between a and b. Note that $n2' = n2 - 1$ for cases (1) and (2); and $n2' = n2 + 1$ for cases (3) and (4). For cases (2) and (3), the reader will accept D' since $D' = D \oplus [I]_0 = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2' \oplus [I]_0 = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2$. For cases (1) and (4), the reader will not accept it due to the mismatch on corresponding (more than one) bit positions. Suppose $n2$ is randomly generated, there is 50% success rate of the simplest attack.

Once the reader accepts the value, the reader needs to update the tag's secret information with the pair $(n1, n2)$. However, the tag uses another pair $(n1, n2')$ to update its secrets. *E.g.*, $IDS_{tag(i)}^{(n+1)} = (IDS_{tag(i)}^{(n)} + (n2' \oplus K4_{tag(i)}^{(n)})) \oplus ID_{tag(i)}$.

It is obvious that there is a mismatch of secret storage for both tag and reader (refer to Table 3). To this end, the simplest attack assumes that there is only one (least significant) bit change on the message C. The attack is efficacious as it will succeed once for two trials. In fact, the simplest attack can be extended to toggle a single bit of C at any location i, so that it can be a general attack with the same (50%) success rate.

Generalized de-synchronization attack: For any on-going LMAP protocol, an adversary can intercept the message C and toggle any bit of C to get C' as $C' = C \oplus [I]_j$ ($0 \leq j \leq 95$). The new message $A||B||C'$ is then sent to the tag. Upon receiving a reply D from the tag, the adversary change it to $D' = D \oplus [I]_j$ and send it to the reader. As per analysis above, the success rate of the attack is 50%. A successful attack may change the tag's secret status on a reader; or say, it de-synchronizes the reader and the tag.

With this attack, some claims in [14] are not true. Noted that the authors introduces an extension LMAP$^+$ in Section 5 of [14], our attack can also be applied on this version directly.

Attack 2: Changing messages A and B. Further on, the attack can also target on $n1$ similarly. In this case, the attacker intercepts the messages $A||B||C$ and sends $A'||B'||C$ to the tag, where $A' = A \oplus [I]_j$ and $B' = B \oplus [I]_j$, i.e., we toggle the j-th bit of A and B. Since $A = IDS_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} \oplus n1$, we set $n1' = n1 \oplus [I]_i$. For B, we obtain

$$If \quad [B]_j = 1; \Rightarrow [B']_j = 0; \rightarrow If \quad [n1]_j = 0, HW(n1 \oplus n1') \geq 2 \qquad (5)$$
$$\rightarrow If \quad [n1]_j = 1, n1' = n1 \oplus [I]_j \qquad (6)$$
$$If \quad [B]_j = 0; \Rightarrow [B']_j = 1; \rightarrow If \quad [n1]_j = 0, n1' = n1 \oplus [I]_j \qquad (7)$$
$$\rightarrow If \quad [n1]_j = 1, HW(n1 \oplus n1'] \geq 2 \qquad (8)$$

in which, $n1' = n1 - 2^j$ for cases (5) and (6); and $n1' = n1 + 2^j$ for cases (7) and (8). For cases (6) and (7), the tag will authenticate the reader by accepting $n1'$. For cases (1) and (4), the tag will not authenticate the reader. Suppose $n1$ is randomly generated, the attacker has 50% success rate to cheat the tag. Suppose the tag accepts the manipulated message (A', B'), it will produce the message D to complete the protocol. The attacker needs to send $D' = D \oplus [I]_j$ to the reader with any valid reply from the tag. And this message D' will be verified by the reader successfully. Upon a successful attack, both reader and tag need to update their secret information. The reader will update with the pair $(n1, n2)$, while the tag uses $(n1', n2)$ that will cause the mismatch in the next execution of authentication protocol (refer to Table 3).

Attack Analysis. Compared with attack 1, where the target is on the partial protocol of the reader authenticating the tag, attack 2 is targeting on the procedure of the tag authenticating the reader. Above attack can be extended to attack 3: if we change $n1$ and $n2$ simultaneously, we do not need to change D anymore. In this case, the attacker intercepts the message and sends $A'||B'||C'$.

Success rate is about 25%. The effects on updating at both the reader and tag side are summarized in Table 3.

Attacks	Success rate	Reader storage	Tag storage
Attack 1	50%	$[IDS, K1, K2, K3, K4]$	$[IDS', K1', K2', K3, K4]$
Attack 2	50%	$[IDS, K1, K2, K3, K4]$	$[IDS, K1, K2, K3', K4']$
Attack 3	25%	$[IDS, K1, K2, K3, K4]$	$[IDS', K1', K2', K3', K4']$

Table 3. Updated storages at the reader and the tag after the attacks

4.2 *Full Disclosure Attack*

Given above attacks, we can further disclose the original ID of a tag, which is much more serious. Suppose 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). That means we can repeatedly run the uncomplete protocol many times at the tag side. The assumption is reasonable as the tag has to answer any request by legitimate or illegitimate readers, and the protocol is not complete if the reader didn't receive final message D.

The attack is illustrated in Fig. 1. Step 1, an attacker impersonates a legitimate reader and gets the current IDS of a tag. Step 2, using this valid IDS the attacker impersonates a tag to get a valid message $A||B||C$ from a legitimate reader. Step 3, the attacker tries to send all possible $A'||B'||C$ to the tag, where A' and B' are obtained by changing the j-th bit of A and B respectively ($0 \leq j \leq 95$). According to whether a proper D or an error message is received (the attacker doesn't need to know the value, an error indicator is enough for an attacker to make his decision), the attacker concludes that the j-th bit of n_1 is equal or not equal to the j-th bit of B. In this way, with merely 96 trials, the attacker can get full bit values of $n1$. Then, from A, B, IDS and $n1$, the attacker can calculate $K1$ and $K2$.

Now, the unknown parameters are $n2, K3, K4$, and ID. Obviously, we can use above method to obtain the value of $n2$, but to interact with the reader m times. However, the repeating trials by the attacker are easily identified by a stateful reader and countered by limiting the interactions by a constant (*e.g.,* up to 10) times. With this assumption, we have to devise another way to derive the secrets. Thus, in Step 4, the attacker pretends to be a legitimate tag and sends the IDS to the readers again (the 2nd interaction with the reader). The reader will response as $A^{new}||B^{new}||C^{new}$. Then, in Step 5, the attacker can set $n1^{new} = 0$ (using the currently known parameters IDS, $K1$ and $K2$) and sends $A^{new'}||B^{new'}||C^{new}$ to the tag. The tag will reply with D^{new}. Note that in the above steps, there are totally 2 interactions between the reader and the attacker, and $m + 2$ interactions between the attacker and the tag.
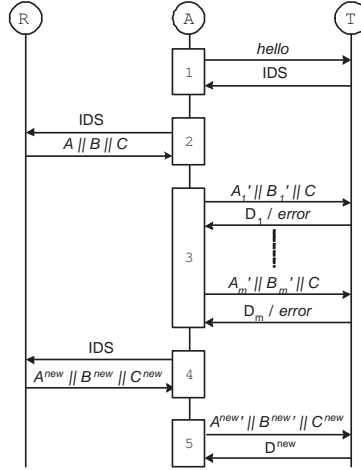
Fig. 1. *Full Disclosure Attack*

To this point, the attacker has the following equations:

$$C = (IDS_{tag(i)}^{(n)} + K3_{tag(i)}^{(n)}) + n2 \tag{9}$$

$$D = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus n2 \tag{10}$$

$$C^{new} = (IDS_{tag(i)}^{(n)} + K3_{tag(i)}^{(n)}) + n2^{new} \tag{11}$$

$$D^{new} = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n2^{new} \tag{12}$$

After that, the attacker can solve the $ID_{tag(i)}$ as follows. First, by eliminating $n2$ from equations (9) and (10), and $n2^{new}$ from equations (11) and (12), we get the equations with unknown parameters ID and $K3$:

$$C - IDS_{tag(i)}^{(n)} - K3_{tag(i)}^{(n)} = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus D \tag{13}$$

$$C^{new} - IDS_{tag(i)}^{(n)} - K3_{tag(i)}^{(n)} = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus D^{new} \tag{14}$$

We further eliminate $K3$ from the above two equations and get

$$C^{new} - C = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus D^{new} - (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1 \oplus D$$

Now, we discuss how to solve $ID_{tag(i)}$ from above equation. Let $a = D^{new}$, $b = n1 \oplus D$, $c = C^{new} - C \mod 2^{96}$, and $x = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \mod 2^{96}$. Since $IDS_{tag(i)}^{(n)}$ is already known, the problem is equivalent to find $x \in \{0,1\}^{96}$ for given (a, b, c) such that

$$x \oplus a = x \oplus b + c \mod 2^{96}. \tag{15}$$

To solve x in equation (15), we just need to note that x's more significant bits do not affect the computation involving its less significant bits. So, we can try to determine x from its less significant bits to its higher significant bits. For example, we can divide the 96 bits into 24 parts so that each part has 4 bits. After that, by exclusively searching we can find all possible solutions for the first 4 less significant bits of x, then the next 4 less significant bits of x, and so on. This procedure involves no more than $(2^{24} - 1)$ times of exclusively searching all 4-bit strings, due to the possible carries at all $(4k + 1)$-th bit locations $(1 \le k \le 23)$. This means that such a naive algorithm can be carried out by a PC in several minutes. Actually, employing efficient algorithms proposed in [10], equation (16) can further be solved in complexity $O(m)$ ($m = 96$ in the protocols). Note that from one given triple (a, b, c), one may not uniquely determine the value of x. In this scenario, the attacker can interact with the reader several times [2] to attain a few instances of equation (15). By intersecting the solution sets of those different instances, the value range of x can be significantly narrowed down. In addition, since $ID_{tag(i)}$ is not a truly random number but has fixed format, some bits of x are almost predefined. Combined with this techniques, it is likely that the value of x can be uniquely determined with enough but not so many interactions with the reader and tag. Once the value of x is fixed, the attack can easily derive the rest secret information ($ID, K3, K4$) stored on the tag. This completes our full disclosure attack against LMAP.

Note that the above full disclosure attack against LMAP protocol can also be adapted for attacking M²AP protocol. Actually, the attack only needs the attacking steps from 1 to 3, from which we can get $n1$. Further on, with a valid E, we obtain ID directly. This implies that the full disclosure attack against M²AP is much more efficient than that on LMAP, since the attacker has only 1 interaction with the reader and $m + 1$ interactions with the tag.

5 Countermeasures

5.1 Re-synchronization

In fact, in a naif extension - LMAP$^+$ of [14], the authors did mentioned a method on re-synchronization between the reader and the tag. The tag will have a state associated in the database: synchronized or uncertainty. Furthermore, each tag will have $l + 1$ database records, instead of only 1 record. The first record is the actual index-pseudonym (IDS) and the others are the potential next index-pseudonyms ($IDS + 1, IDS + 2, ..., IDS + l$). The parameter l is decided by the size of the database, thus it can not be too large for all records being stored in a database. The extension can help re-synchronize some situations

---

[2] Not necessarily in a single protocol run, that means the attacker can launch the attack for an arbitrary protocol run, perhaps after several successful protocol executions, and no matter how many times the tag has updated its secret information. Even a stateful reader is not able to detect the subtle attack.

of asynchronization. Unfortunately, the method can only affect only a small percent on the efficacy of our attack. Suppose $l \leq 2^L$, for all $[I]_i, (L < i \leq m)$, our attack can still succeed with 95% [3] trials. One natural remedy against the *de-synchronization attack* is to build bit level error correcting mechanisms at the database. However, the bit errors between mismatched IDSs as well as other secrets, like $K1, K2, K3, K4$, are not easily corrected in this way. Since the combination usage of bitwise operations (*e.g.*, $\oplus, + \bmod 2^m$) broke the algebraic property of their functions. A single bit flip on $n1$ or $n2$ may cause different bit error patterns on updated secret values, where an adaptive error correction mechanism should be deployed. Hence, additional costs on computation and storage at the database are incurred.

5.2 Sending $\widetilde{D}$

One of the trick we did in our *Full-disclosure attack* is to try all possible $A'||B'||C$ and observe the replies from the tag (in step 3). If the tag sends a valid message D, that means the trial is successful; if not, the trial is not successful. Suppose the tag always sends a message $\widetilde{D}$ implicitly whatever the reader is authenticated or not ($\widetilde{D} = D$, if the reader is authenticated; Or $\tilde{D} \in_R \{0,1\}^m$, if the reader is not authenticated). Then, the attacker can not get any clue on distinguishing a valid message D or an arbitrary message. The attacker has to send it to a legitimate reader and expects a reply. As it might not be possible for a tag to generate a random value $\{0,1\}^m$, the tag can assign $\widetilde{D} = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n2$, if the reader is not authenticated. As long as $\widetilde{D}$ is distinguishable for the reader and indistinguishable for the attacker, any other (secure) mechanism will work.

5.3 Storing Status

Storing some addendum at the database alone might not be helpful, but storing some status information at both the reader and the tag sides could be useful to counter our attacks. The intuition is that our attack targets on the tag's inability to distinguish the requests from a legitimate reader or the trials from an attacker. To counter our attack, it is necessary for a tag to have some status information stored, to indicate the trials of some on-going sessions. To this end, we assign an additional status bit s, and set $s = 0$, if the protocol is completed (or synchronized) successfully; or $s = 1$, if the protocol is uncompleted (or asynchronized) due to some reason.

The protocol status bit is set for the purpose of indicating the completion of a protocol execution. Only a successful completed protocol can trigger the updating operations at both the reader and the tag sides. That means an attacker

---

[3] The success rate is about $(m - \log_2 l)/m$, for all $(0 \leq l \leq 2^m)$. Set $l = 32, m = 96$, we get $91/96 \approx 95\%$.

can not learn the bit values of $n1$ or $n2$ within a single (incomplete) protocol by launching multiple (failed) trials.

Therefore, the stateful protocol needs both the reader and the tag to store two random numbers in the last (incomplete) protocol round $(n1, n2)$, in case of asynchronization. Given an incomplete protocol, the tag will expect a completion message E (e.g., $E = (IDS_{tag(i)}^{(n+1)} + ID_{tag(i)}) \oplus n1 \oplus n2$) from the reader. The reader, already updated, needs to search the database to calculate a former IDS of the tag with the stored values $(n1, n2)$. If a former IDS is found, the reader further composes the completion message E and sends it to the tag to complete the protocol. If not, a tag is considered compromised permanently.

With only 1 bit added to present the protocol status and two additional random numbers $(2 * 96 = 192$ bits$)$ stored in EEPROM, the new protocol increases a tag's memory size by 40% $(193/96 * 5)$, while nearly all other hardware implementations for algorithm logic units or control units are not changed.

Above we proposed several countermeasures against different attacks. While there must be some other ways on attacking the protocols, the current countermeasures might not guarantee the security due to attacks discovered later on. Some of the countermeasures can be combined to provide stronger security for the protocols. However, the new mechanisms have some limitations to be deployed in some real situations. For example, the stateful protocol is not suitable for ubiquitous environment, where distributed readers can not retrieve the tags' information efficiently online so as to authenticate a tag.

6 Conclusions and Future Work

In this paper, we demonstrated two effective attacks against two ultra-lightweight RFID mutual authentication protocols, which are recently proposed in [14, 15]. The severity of the attacks indicates the insecure design of the protocols. Our work shows that it may be quite dangerous on using only simple bitwise operations to achieve secure RFID mutual authentication under powerful adversarial model. The security of such protocols must be proved with elaborated cryptanalysis. To counter those attacks, some countermeasures were also presented to deal with disruptive attacks. Taken these attacks and countermeasures in mind, our next step is to design secure (ultra) lightweight RFID mutual authentication protocol and to apply it on low-cost RFID tags.

7 Acknowledgement

References

1. M. Aigner and M. Feldhofer. Secure Symmetric Authentication for RFID Tags. *Telecommunication and Mobile Computing*, March 2005.
2. G. Avoine. Security and Privacy in RFID Systems. `http://lasecwww.epfl.ch/~gavoine/rfid/`
3. J. Bringer, H. Chabanne, and E. Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In: *Proc. of SecPerU'06*, pp. 28-33. IEEE Computer Society Press, 2006.
4. B. Defend, K. Fu and A. Juels. Cryptanalysis of Two Lightweight RFID Authentication Schemes. In: *Proc. of PERSEC'07*, March 2007.
5. S. Garfinkel and B. Rosenberg. RFID: Applications, Security, and Privacy. Published by *Addison-Wesley Professional*, 2005.
6. A. Juels and R. Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In: *Proc. of FC'03*, LNCS 2742, pp. 103-121. Springer-Verlag, 2003.
7. A. Juels. Minimalist Cryptography for Low-Cost RFID Tags. In: *Proc. of SCN'04*, LNCS 3352, pp. 149-164. Springer-Verlag, 2004.
8. A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In: *Proc. of CRYPTO'05*, LNCS 3126, pp. 293-308. Springer-Verlag, 2005.
9. A. Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2): 381-394, Feb. 2006.
10. H. Lipmaa and S. Moriai. Efficient Algorithms for Computing Differential Properties of Addition. In: *Proc. of FSE '01*, LNCS 2355, pp. 336-350. Springer-Verlag, 2001.
11. D. Molnar and D. Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: *Proc. of CCS'04*, pp. 210-219. ACM Press, 2004.
12. D. Molnar, A. Soppera, and D. Wagner. A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In: *Proc. of SAC'05*, LNCS 3897, pp. 276-290. Springer-Verlag, 2005.
13. M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to privacy-friendly tags. In: *Proc. of RFID Privacy Workshop*, 2003.
14. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags. In: *Proc. of 2nd Workshop on RFID Security*, July 2006. `http://events.iaik.tugraz.at/RFIDSec06/`.
15. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. M^2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags. In: *Proc. of International Conference on Ubiquitous Intelligence and Computing UIC'06*, LNCS 4159, pp. 912-923. Springer-Verlag, 2006.
16. I. Vajda and L. Buttyan. Lightweight authentication protocols for low-cost RFID tags. In: *Proc. of UBICOMP'03*, 2003.
17. S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In: *Proc. of 1st Int. Conf. on Security in Pervasive Computing*, LNCS 2802, pp. 201-212. Springer-Verlag, 2003.
18. S. Weis. Security parallels between people and pervasive devices. In: *Proc. of PERSEC'05*, pp. 105-109. IEEE Computer Society Press, 2005.