Security Analysis of $LMAP^{++}$, an RFID Authentication Protocol

Nasour Bagheri
E.E. Dept., SRTTU,
Tehran, Iran, 16788-15811,
Tel/fax:+98-21-2297006,
Email:nbagheri@srttu.edu

Masoumeh Safkhani
E.E. Dept.,
IUST,
Tehran, Iran
Email: m_safkhani@iust.ac.ir

Majid Naderi
E.E. Dept.,
IUST,
Tehran, Iran,
Email: m_naderi@iust.ac.ir

Somitra Kumar Sanadhya
Indraprastha Institute of
Information Technology (IIIT)
Delhi, New Delhi, India
Email: Somitra@iiitd.ac.in

***Abstract*—Low cost RFID tags are increasingly being deployed in various practical applications these days. Security analysis of the way these tags are used in an application is a must for successful adoption of the RFID technology. Depending on the requirements of the particular application, security demands on these tags cover some or all of the aspects such as privacy, untraceability and authentication. As a result of increasing deployment of RFID tags, many works on RFID protocols and their security analysis have appeared in the literature in the past few years. Although most protocol proposals also provide some justification for the claimed security properties of these protocols, independent third party evaluation has often revealed weaknesses in these protocols. In this work, we present a third party security evaluation of a recently proposed mutual authentication protocol $LMAP^{++}$.**

Mutual authentication protocols are an important class of protocols for RFID applications. In these protocols, the reader and the tag of an RFID system run an interactive game to authenticate themselves to each other. In this work, we present traceability and desynchronization attacks against the protocol $LMAP^{++}$. First we show that $LMAP^{++}$ does not satisfy the security notion of traceability as defined in the model proposed by Jules and Weis. Using the ideas of this traceability attack, next we show that $LMAP^{++}$ also suffers from a desynchronization attack. The presented attacks have low complexities and high success probabilities. To the best of our knowledge, this the first attack on the $LMAP^{++}$ protocol.

***Keywords*-Desynchronization, $LMAP^{++}$, Mutual Authentication Protocol, Privacy, RFID, Traceability.**

I. Introduction

Radio Frequency Identification (RFID) technology is a wireless identification method that uses radio frequency to send and receive data. Most of the RFID systems comprise of three entities: the tag, the reader and the back-end database. The tag is a highly constrained microchip with antenna that stores the unique tag identifier and other related information about an object that the tag has been attached to. The reader is a device that can read/modify the stored information of the tags and (if needed) transfer these data to a back-end database, with or without modification. In general, the reader stores tags identifiers, pseudonyms and secrets in the back-end database. In addition, the back-end database is usually not resource constrained and has the ability to carry out more complex calculations.

RFID technology is finding more and more applications in modern life. For instance, this technology is being used in national passports, retail goods in supermarkets and travel cards among others. The security analysis of RFID protocols, specially third party analysis, is crucial to ensure that these ubiquitous uses of the technology remain secure. One of the principal security aspects of an RFID system is authentication. Mutual authentication protocols are used to securely authenticate tags and readers to each other. Several lightweight mutual authentication protocols proposed in the literature [1], [2], [3], [4], [5] have already have been broken [6], [7], [8], [9], [10], [11], [12], [13].

In [14] Peris *et al.* proposed a lightweight mutual authentication protocol called $LMAP$. In addition, they proposed an extension of this protocol and called it $LMAP^{+}$. These protocols are extremely lightweight and use only simple bitwise operations. However, it has been discovered very soon that these protocols do not achieve the claimed security [15]. Later, following the $LMAP$ designing strategy, Li [16] proposed a new lightweight protocol. Li [16] also called the proposed scheme $LMAP^{+}$. However, to avoid confusion with the extension of $LMAP$ proposed by Peris *et al.* in [14], we call Li's scheme $LMAP^{++}$ protocol in the rest of this paper. The $LMAP^{++}$ protocol can be seen as a modified version of $SLMAP$ protocol [17] which has been analyzed in [18], [19].

In this work we investigate the security of the $LMAP^{++}$ protocol and present two attacks for this protocol. More precisely, we show that this protocol does not satisfy the security notion of traceability as defined by Jules and Weis [20], which has been later used by Phan in his attack against SASI [13]. This can be seen as a traceability attack on this protocol which has the success probability of '1' and can be performed in one run of protocol. In addition, we present a desynchronization attack against the $LMAP^{++}$ protocol which has the success probability of 2^{-4} on each run of protocol.

The rest of the paper is organized as follows: Notation is introduced in Section II and Section III describes the $LMAP^{++}$ protocol. Our traceability attack is presented in Section IV. Section V explains our desynchronization attack. Finally, we conclude with suggestions for improving

$LMAP^{++}$ in section VI.

II. Notation

The notations used in this paper are as follows:

- $ID_{tag(i)}$: indicates tag's static identifier.
- $PID_{tag(i)}^{(n)}$: indicates tag's dynamic pseudonym at the n^{th} successful run of protocol.
- $K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$: indicate tag's secret keys at the n^{th} successful run of protocol.
- r: indicates a pseudorandom number which is generated by the reader.
- $\oplus$: indicates XOR operation.
- $\|$: indicates concatenation operator.
- $+$: indicates addition mode 2^m.
- All parameters in the protocol are of length 96-bit.
- The expression $A \rightarrow B$ refers to assigning A to B.
- For a finite set $\mathcal{X}$, $x \xleftarrow{\$} \mathcal{X}$ is the experiment of uniformly choosing a random element from $\mathcal{X}$ and assigning it to x.
- The n^{th} bit of X is denoted by $(X)_n$. Hence, the least significant bit(LSB) of X is denoted by $(X)_0$ (similarly for the most significant bit).

III. $LMAP^{++}$ Description

In the $LMAP^{++}$ protocol, each tag has a static identifier. The identifier of the i^{th} tag is indicated by $ID_{tag(i)}$. In addition, each tag has a pseudonym $PID_{tag(i)}$ and shares two secret keys i.e. $K1_{tag(i)}$ and $K2_{tag(i)}$ which get updated after each successful run of the protocol. We denote the values of $PID_{tag(i)}$, $K1_{tag(i)}$ and $K2_{tag(i)}$ at the n^{th} successful run of protocol by $PID_{tag(i)}^{(n)}$, $K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$ respectively. Hence, in this protocol, the tag and the reader save the tuple $(ID_{tag(i)}, PID_{tag(i)}^{(n)}, K1_{tag(i)}^{(n)}, K2_{tag(i)}^{(n)})$. We denote a table that the reader stores these tuples into by T_T. This table is indexed by the $PID_{tag(i)}^{(n)}$ values. On receiving a $PID_{tag(i)}^{(n)}$ from a tag, the reader looks into T_T. If $PID_{tag(i)}^{(n)} \in T_T$ the reader extracts the related $K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$ and continues the game. Otherwise, the reader terminates the game.

To initiate a mutual authentication session, the reader will send a "hello" to the tag. The Tag answers by sending its current pseudonym $PID_{tag(i)}$. The reader looks up into T_T for this PID. If $PID_{tag(i)}^{(n)} \in T_T$, the reader extracts the related $K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$ and combines them with a random value r to generate A and B as follows:

$$A \leftarrow PID_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} + r$$

$$B \leftarrow PID_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} \oplus r$$

Next, the reader passes $A\|B$ to the tag. The tag extracts r from A and uses it together with B to authenticate the reader. If the Tag authenticates the reader, it calculates a new variable C, passes it to the reader and updates $PID_{tag(i)}^{(n)}, K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$ as follows:

$$C \leftarrow (PID_{tag(i)}^{(n)} + ID_{tag(i)} \oplus r) \oplus (K1_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} + r)$$

The reader verifies the received C to authenticate the tag and updates $PID_{tag(i)}^{(n)}, K1_{tag(i)}^{(n)}$ and $K2_{tag(i)}^{(n)}$. The details of $LMAP^{++}$ are depicted in Algorithm 1. It should be mentioned that all parameters of algorithm are of length $n = 96$ bits.

To overcome the desynchronization attacks, the protocol designer has considered a status bit in the protocol denoted by s. In each run, if the protocol successfully completed, s will be initialized with 0 otherwise it sets to 1. Hence, $s = 1$ indicates that the protocol was not successfully completed. However, this bit has no affect on our attacks.

IV. Traceability Attack

Our traceability attack follows the model for traceability proposed by Jules and Weis in [21]. This model of traceability has later been used by Phan [13] in their attack against SASI[3]. In this traceability model, the attacker is given the static identifiers of two distinct tags, e.g. T_0 and T_1, and participates in a game of one successful run of the protocol with one of these two identifiers. The attacker has to predict which tag is being used. Now, if the attacker can guess which tag has been involved in the game correctly it wins and we say the protocol suffers from traceability attack. The adversary makes its decision public by output a bit, namely "0" for T_0 and "1" for T_1. The attacker succeeds on the distinguishing between tags if the probability of his correct guess has a non-negligible derivation from the random guess probability, 0.5. In other words, given the statics ID of T_0 and T_1, i.e. ID_0 and ID_1, the adversary's advantage, Adv_A, on mounting the traceability attack on the protocol is given as follows:

$$Avd_A(ID_0, ID_1) = |Pr_{CG} - Pr_{RG}| = \left| Pr_{CG} - \frac{1}{2} \right|$$

where, Pr_{CG} and Pr_{RG} indicate the probabilities of correct guess and random guess respectively. Following the above model, we propose a traceability attack on $LMAP^{++}$ which has been depicted in Algorithm 2. In this attack, we assumed that $(ID_0)_0 = 0$ and $(ID_1)_0 = 1$. The attack includes two phases, the Online phase and the Offline phase. In the Online phase the adversary eavesdrops all transferred messages of one run of protocol. In the Offline phase of attack, the adversary uses the fact that considering only the last significant bit(LSB) modular additions mod 2^m can be replaced by bitwise XOR. Hence, based on the protocol construction depicted on Algorithm 1,we can write the following equalities:

$$(A)_0 = (PID_{tag(i)}^{(n)})_0 \oplus (K1_{tag(i)}^{(n)})_0 \oplus (r)_0$$

$$(B)_0 = (PID^{(n)}_{tag(i)})_0 \oplus (K2^{(n)}_{tag(i)})_0 \oplus (r)_0$$

$$(C)_0 = (PID^{(n)}_{tag(i)})_0 \oplus (ID_{tag(i)})_0 \oplus$$

$$(r)_0 \oplus (K1^{(n)}_{tag(i)})_0 \oplus (K2^{(n)}_{tag(i)})_0 \oplus (r)_0$$

Therefore, the adversary can eavesdrop one successful run of protocol, store $PID^{n}_{tag(i)}$, A, B and C and extracts $(ID_{tag(i)})_0 \in (0,1)$ as follows:

$$(ID_{tag(i)})_0 \leftarrow (A)_0 \oplus (B)_0 \oplus (C)_0 \oplus$$

$$(PID^{(n)}_{tag(i)})_0 \oplus (PID^{(n)}_{tag(i)})_0 \oplus (PID^{(n)}_{tag(i)})_0$$

Hence, following the assumption that $(ID_0)_0 = 0$ and $(ID_1)_0 = 1$, the adversary can distinguish with the probability of '1' whether he is interacting with T_0 or T_1.

V. Desynchronization Attack

In this section we present a desynchronization attack against the $LMAP^{++}$ protocol. The main technique is to force the tag and the reader to update their common values to different numbers. If the adversary can succeed in forcing the tag and the reader to do so, they will not authenticate each other in further transactions.

Our desynchronization attack on $LMAP^{++}$ is based on an assumption that $(PID^{(n)}_{tag(i)})_0$, $(K1^{(n)}_{tag(i)})_0$, $(K2^{(n)}_{tag(i)})_0$ and $(ID)_0$ are zero. To mount the attack, the adversary eavesdrops a transferred value $A\|B$ from the reader to the tag and toggles the LSB bits of A and B, $(A)_0$ and $(B)_0$. Considering the above assumption on $(PID^{(n)}_{tag(i)})_0$, $(K1^{(n)}_{tag(i)})_0$ and $(K2^{(n)}_{tag(i)})_0$, the carry of modular addition will not propagated from the lowest significant bit to the next bit. In addition, modular addition for LSBs can be replaced by exclusive or. Hence, we have:

$$(A)_0 \leftarrow (PID^{(n)}_{tag(i)})_0 \oplus (K1^{(n)}_{tag(i)})_0 \oplus (r)_0$$

$$(B)_0 \leftarrow (PID^{(n)}_{tag(i)})_0 \oplus (K2^{(n)}_{tag(i)})_0 \oplus (r)_0$$

So, if we toggle the LSBs of r, A and B it has no impact on the correctness of the above equations and the tag authenticates the reader with the probability of '1'. However, the extracted random value by the tag, r', does not equal to what is generated by the reader, r, and we have $r' = r \oplus 1$. On the other hand, in the next step of the protocol, the tag passes C to the reader which is calculated as follows:

$$C \leftarrow (PID^{(n)}_{tag(i)} + ID_{tag(i)} \oplus r') \oplus (K1^{(n)}_{tag(i)} + K2^{(n)}_{tag(i)} + r')$$

Considering the assumption that $(PID^{(n)}_{tag(i)})_0$, $(K1^{(n)}_{tag(i)})_0$, $(K2^{(n)}_{tag(i)})_0$ and $(ID)_0$ are zero, replacing r by $r' = r \oplus 1$ has no affect on the generated value for C because considering the calculation for $(C)_0$ we have:

$$(C)_0 = ((PID^{(n)}_{tag(i)})_0 + (ID_{tag(i)})_0 \oplus (r)_0) \oplus$$

$$((K1^{(n)}_{tag(i)})_0 + (K2^{(n)}_{tag(i)})_0 + (r)_0) =$$

$$(0 + 0 \oplus (r)_0) \oplus (0 + 0 + (r)_0) = (r)_0 \oplus (r)_0 =$$

$$((r)_0 \oplus 1) \oplus ((r)_0 \oplus 1) = (r')_0 \oplus (r')_0 =$$

$$((PID^{(n)}_{tag(i)})_0 + (ID_{tag(i)})_0 \oplus (r')_0) \oplus$$

$$((K1^{(n)}_{tag(i)})_0 + (K2^{(n)}_{tag(i)})_0 + (r')_0)$$

In addition, carry will not propagated from $(C)_0$ to $(C)_1$ neither with r nor r'. Hence, the reader also authenticates the tag with the probability of '1' and both the tag and the reader update the values of $PID^{(n)}_{tag(i)}$, $K1^{(n)}_{tag(i)}$ and $K2^{(n)}_{tag(i)}$. However, the tag uses $r' = r \oplus 1$ in updating phase of protocol while the reader uses r. Thereby, the tag exits from synchronism with the reader and the tag and the reader can not authenticate each other in any following runs of the protocol.

To determine the success probability of the attack, we can combine the success probabilities of each stage of the above attack. At the beginning of attack we assumed that $(PID^{(n)}_{tag(i)})_0$, $(K1^{(n)}_{tag(i)})_0$,$(K2^{(n)}_{tag(i)})_0$ and $(ID)_0$ are zero. This assumption could be valid with the probability of $\frac{1}{2^4}$. If the above assumption is correct then the success probability of the rest of attack would be '1'. Hence, we can conclude that the total success probability of attack is $\frac{1}{2^4}$. Therefore, if $(ID)_0 \neq 0$, the attacker can repeat the attack a few times to desynchronize the tag and the reader. The details of the attack are depicted in Algorithm 3.

VI. Conclusion

In this paper we consider the security of one of the recently proposed lightweight RFID authentication protocol $LMAP^{++}$, which is a successor of the $LMAP$ and $LMAP^{+}$ protocols. In this paper we presented traceability and desynchronization attacks against this protocol. Our traceability attack has a negligible complexity and the complexity of the proposed desynchronization attack is a few runs of protocol.

To fix the above vulnerability it should be enough to use rotation on the computation of the communicated messages, A, B and C. In this way, the adversary may not apply the attacks presented in this work. However, our results and previous attacks on other authentication protocols that have not employed any cryptographic primitives, e.g. SASI, have shown that it would not be an easy task to design a secure protocol based on this strategy. Hence, we prefer to not introduce any concrete variant for this protocol. Designing a lightweight RFID mutual authentication protocol which does not suffer from attacks of the kind presented in this paper is a challenging problem.

REFERENCES

[1] Alireza Sadighian and Rasoul Jalili. Afmap: Anonymous forward-secure mutual authentication protocols for rfid systems. In *The Third IEEE International Conference on Emerging Security Information, Systems and Technologies(SECURWARE 2009)*, pages 31–36, 2009.

[2] Alireza Sadighian and Rasoul Jalili. Flmap: A fast lightweight mutual authentication protocol for rfid systems. In *The 16th IEEE International Conference On Networks (ICON 2008)*, pages 1–6, New Delhi, India, 2008.

[3] Hung-Yu Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, December 2007.

[4] Pedro Peris-Lopez, Julio César Hernández Castro, Juan M. Estévez-Tapiador, and Arturo Ribagorda. Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol. In *WISA*, pages 56–68, 2008.

[5] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagorda. EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags. In *OTM Federated Conferences and Workshop: IS Workshop – IS'06*, volume 4277 of *Lecture Notes in Computer Science*, pages 352–361, Montpellier, France, November 2006. Springer-Verlag.

[6] Masoumeh Safkhani, Majid Naderi, and Nasour Bagheri. Cryptanalysis of AFMAP. *IEICE Electronics Express*, 7(17):1240–1245, 2010.

[7] Masoumeh Safkhani, Majid Naderi, and Habib Rashvand. Cryptanalysis of AFMAP. *International Journal of Computer & Communication Technologys*, 2(2):182–186, 2010.

[8] Mihály Bárász, Balázs Boros, Péter Ligeti, Krisztina Lója, and Dániel Nagy. Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[9] Tianjie Cao, Elisa Bertino, and Hong Lei. Security analysis of the sasi protocol. *IEEE Transactions on Dependable and secure Computing*, 6(1):73–77, 2009.

[10] Julio C Hernandez-Castro, Juan M E Tapiador, Pedro Peris-Lopez, and Jean-Jacques Quisquater. Cryptanalysis of the sasi ultralightweight rfid authentication protocol with modular rotations. Technical Report arXiv:0811.4257, Nov 2008.

[11] Tieyan Li and Robert H. Deng. Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol. In *Second International Conference on Availability, Reliability and Security – AReS 2007*, Vienna, Austria, April 2007.

[12] Tieyan Li, Guilin Wang, and Robert H. Deng. Security Analysis on a Family of Ultra-lightweight RFID Authentication Protocols. *Journal of Software*, 3(3), March 2008.

[13] Raphael C.-W. Phan. Cryptanalysis of a new ultralightweight rfid authentication protocol - sasi. *IEEE Transactions on Dependable and Secure Computing*, 6(4):316–320, 2009.

[14] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagorda. Lmap: A real lightwight mutual authentication protocol for low-cost rfid tags. In *Proceedings of RFIDSec06 Workshop on RFID Security*, Graz,Austria, 12-14 July 2006.

[15] Tieyan Li and Guilin Wang. Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols. In *IFIP SEC 2007*, Sandton, Gauteng, South Africa, May 2007.

[16] Tieyan Li. Employing lightweight primitives on low-cost rfid tags for authentication. In *VTC Fall*, pages 1–5, 2008.

[17] Tieyan Li and Guilin Wang. SLMAP - a secure ultra-lightweight RFID mutual authentication protocol. In *Chinacrypt07*, pages 19–22, 2007.

[18] Julio C. Hernandez-Castro, Juan E. Tapiador, Pedro Peris-Lopez, John A. Clark, and El-Ghazali Talbi. Metaheuristic traceability attack against SLMAP, an RFID lightweight authentication protocol. In *Workshop on Nature Inspired Distributed Computing, 23rd IEEE International Symposium on Parallel and Distributed Processing (23rd IPDPS'09)*, pages 1–5. IEEE, 2009.

[19] Julio C. Hernandez-Castro, Juan E. Tapiador, Pedro Peris-Lopez, John A. Clark, and El-Ghazali Talbi. Metaheuristic traceability attack against SLMAP, an RFID lightweight authentication protocol. *Int. J. Foundations of Computer Science*, To appear.

[20] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *PerCom Workshops*, pages 342–347. IEEE Computer Society, 2007.

[21] A Jules and S A Weis. Defining strong privacy for rfid. In *Proceedings of IEEE PerCom'07*, pages 342–347, 2007.

The reader;
Sends a $Hello$ message to the tag;

The tag;
Passes its pseudonym $PID_{tag(i)}^{(n)}$ to the reader;
The reader;
if $PID_{tag(i)}^{(n)} \in T_T$ **then**
  $r \xleftarrow{\$} \{0,1\}^t$;
  $A \leftarrow PID_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} + r$;
  $B \leftarrow PID_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} \oplus r$;
  Passes $A||B$ to the tag;
else
  The protocol will be terminated;
end

The tag;
$r_1 \leftarrow A - (PID_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)})$; // Extracting r from A;
$r_2 \leftarrow (B - PID_{tag(i)}^{(n)}) \oplus K2_{tag(i)}^{(n)}$; // Extracting r from B;
if $r_1 = r_2$ **then**
  The tag authenticates the reader;
  $C \leftarrow (PID_{tag(i)}^{(n)} + ID_{tag(i)} \oplus r) \oplus (K1_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} + r)$;
  Passes C to the reader;
  $PID_{tag(i)}^{(n+1)} \leftarrow (PID_{tag(i)}^{(n)} + K1_{tag(i)}^{(n)}) \oplus r + (ID_{tag(i)} + K2_{tag(i)}^{(n)}) \oplus r$; // Updating the PID value ;
  $K1_{tag(i)}^{(n+1)} \leftarrow K1_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K2_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K1$ value ;
  $K2_{tag(i)}^{(n+1)} \leftarrow K2_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K1_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K2$ value ;
else
  The tag does not authenticate the reader;
  $C \xleftarrow{\$} \{0,1\}^t$;
  Outputs C;
end

The reader;
$C^* \leftarrow (PID_{tag(i)}^{(n)} + ID_{tag(i)} \oplus r) \oplus (K1_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} + r)$;
if $C = C^*$ **then**
  The reader authenticates the tag;
  $PID_{tag(i)}^{(n+1)} \leftarrow (PID_{tag(i)}^{(n)} + K1_{tag(i)}^{(n)}) \oplus r + (ID_{tag(i)} + K2_{tag(i)}^{(n)}) \oplus r$; // Updating the PID value ;
  $K1_{tag(i)}^{(n+1)} \leftarrow K1_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K2_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K1$ value ;
  $K2_{tag(i)}^{(n+1)} \leftarrow K2_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K1_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K2$ value ;
else
  The reader does not authenticate the tag;
end

Algorithm 1. The $LMAP^{++}$ description on round n.

Online Phase;
Eavesdrop one successful run of protocol and store $PID^{n}_{tag(i)}$, A, B and C;
Offline Phase;
Extract $(ID_{tag(i)})_0 \in (0,1)$ as follows;
$(A)_0 \leftarrow (PID^{(n)}_{tag(i)})_0 \oplus (K1^{(n)}_{tag(i)})_0 \oplus (r)_0$;
$(B)_0 \leftarrow (PID^{(n)}_{tag(i)})_0 \oplus (K2^{(n)}_{tag(i)})_0 \oplus (r)_0$;
$(C)_0 \leftarrow (PID^{(n)}_{tag(i)})_0 \oplus (ID_{tag(i)})_0 \oplus (r)_0 \oplus (K1^{(n)}_{tag(i)})_0 \oplus (K2^{(n)}_{tag(i)})_0 \oplus (r)_0$;
$(ID_{tag(i)})_0 \leftarrow (A)_0 \oplus (B)_0 \oplus (C)_0 \oplus (PID^{(n)}_{tag(i)})_0 \oplus (PID^{(n)}_{tag(i)})_0 \oplus (PID^{(n)}_{tag(i)})_0$;
// $(ID_{tag(i)})_0 \in (0,1)$ that simply distinguishes between T_0 and T_1;
Decide the game as follows:
if $(ID_{tag(i)})_0 = 0$ **then**
  Output "0";
else
  Output "1";
end

Algorithm 2. The Traceability Attack Against $LMAP^{++}$.

The reader;
Sends a $Hello$ message to the tag;

The tag;
Passes its pseudonym $PID_{tag(i)}^{(n)}$ to the reader;
The reader;
if $PID_{tag(i)}^{(n)} \in T_T$ **then**
  $r \xleftarrow{\$} \{0,1\}^t$;
  $A \leftarrow PID_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} + r$;
  $B \leftarrow PID_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} \oplus r$;
  Passes $A||B$ to the tag;
else
  The protocol will be terminated;
end

The Attacher;
eavesdrops A and B;
$A \leftarrow A \oplus 1$; // toggling the LSB of A;
$B \leftarrow B \oplus 1$; // toggling the LSB of B;
Passes $A||B$ to the tag;

The tag;
$r_1 \leftarrow A - (PID_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)})$; // Extracting r from A. It can be seen that $r_1 = r + 1$;
$r_2 \leftarrow (B - PID_{tag(i)}^{(n)}) \oplus K2_{tag(i)}^{(n)}$; // Extracting r from B. It can be seen that $r_2 = r + 1$;
if $r_1 = r_2$ **then**
  The tag authenticates the reader; // $r_1 = r_2 = r + 1$, hence the tag authenticates the reader;
  $C \leftarrow (PID_{tag(i)}^{(n)} + ID_{tag(i)} \oplus r') \oplus (K1_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} + r')$; // $r' = r + 1$;
  Passes C to the reader;
  $PID_{tag(i)}^{(n+1)} \leftarrow (PID_{tag(i)}^{(n)} + K1_{tag(i)}^{(n)}) \oplus r' + (ID_{tag(i)} + K2_{tag(i)}^{(n)}) \oplus r'$; // Updating the PID value ;
  $K1_{tag(i)}^{(n+1)} \leftarrow K1_{tag(i)}^{(n)} \oplus r' + (PID_{tag(i)}^{(n+1)} + K2_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K1$ value ;
  $K2_{tag(i)}^{(n+1)} \leftarrow K2_{tag(i)}^{(n)} \oplus r' + (PID_{tag(i)}^{(n+1)} + K1_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K2$ value ;
else
  The tag does not authenticate the reader;
  $C \xleftarrow{\$} \{0,1\}^t$;
  Outputs C;
end

The reader;
$C^* \leftarrow (PID_{tag(i)}^{(n)} + ID_{tag(i)} \oplus r) \oplus (K1_{tag(i)}^{(n)} + K2_{tag(i)}^{(n)} + r)$;
if $C = C^*$ **then**
  The reader authenticates the tag;
  $PID_{tag(i)}^{(n+1)} \leftarrow (PID_{tag(i)}^{(n)} + K1_{tag(i)}^{(n)}) \oplus r + (ID_{tag(i)} + K2_{tag(i)}^{(n)}) \oplus r$; // Updating the PID value ;
  $K1_{tag(i)}^{(n+1)} \leftarrow K1_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K2_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K1$ value ;
  $K2_{tag(i)}^{(n+1)} \leftarrow K2_{tag(i)}^{(n)} \oplus r + (PID_{tag(i)}^{(n+1)} + K1_{tag(i)}^{(n)} + ID_{tag(i)})$; // Updating the $K2$ value ;
else
  The reader does not authenticate the tag;
end

Algorithm 3. The Desynchronization Attack against $LMAP^{++}$.