International Journal of Computer and Communication Technology

Volume 3 | Issue 2 Article 10

April 2012

Cryptanalysis of the Fast Lightweight Mutual Authentication Protocol

Masoumeh Safkhani
Iran University of Science and Technology, Tehran, Iran1,2, m_Safkhani@iust.ac.ir

Majid Naderi
Iran University of Science and Technology, Tehran, Iran1,2, M_Naderi@iust.ac.ir

Habib F. Rashvand
Electrical Engineering Department, School of Engineering , University of Warwick3, H.Rashvand@warwick.ac.uk

Recommended Citation

Safkhani, Masoumeh; Naderi, Majid; and Rashvand, Habib F. (2012) "Cryptanalysis of the Fast Lightweight Mutual Authentication Protocol," *International Journal of Computer and Communication Technology*: Vol. 3 : Iss. 2 , Article 10.
DOI: 10.47893/IJCCT.2012.1127
Available at: https://www.interscience.in/ijcct/vol3/iss2/10

Cryptanalysis of the Fast Lightweight Mutual Authentication Protocol

Masoumeh Safkhani [1], Majid Naderi [2], Habib F. Rashvand[3]
Electrical Engineering Department, School of Engineering , University of Warwick[3]
Iran University of Science and Technology, Tehran, Iran[1,2,] United Kingdom[3]
m_Safkhani@iust.ac.ir[1], M _Naderi@iust.ac.ir [2], H.Rashvand@warwick.ac.uk[3]

Abstract*- Security counts as a critical barrier to continuing growth of RFID industry due to lack of a proper high performance lightweight protocol-based solution. Amongst recent developments the Fast Lightweight Mutual Authentication Protocol (FLMAP) has been accepted for its superior speed and low complexity features. Here we examine the security strengths of FLMAP through systematic cryptanalysis tests. Outcome of our investigation show that in spite of its superior speed and power saving features FLMAP shows some serious design gaps and shortfalls against two specifically selected desynchronization and ID disclosure attacks. *Finally, we propose solutions to fix the FLMAP designing and security flaws.

Keywords-Security Test, RFID, FLMAP, Desynchronization, ID Disclosure, Cryptanalysis

1. INTRODUCTION

Radio Frequency Identification (RFID) is a wireless tag identification system that incorporates three entities: the tag, the reader and the back-end database. The tag is a highly constraint microchip equipped with an antenna that stores the identifier and other related information about the holder of the tag. The reader is a device that can read or modify the stored information in the tags and transfer them to a back-end database, with or without modification. The reader stores tags identifiers, pseudonyms and secrets in the back-end database.

Upon recent growth of RFID applications due to accommodating private information and associated implications on human-life many have raised serious security concerns for their unprotected use putting extra pressure on these devices. Most important is that RFID systems normally need to operate over limited resources and work under very restricted conditions where only light weight security protocols can be adopted. Many mutual authentication protocols are proposed so that tags and readers can securely authenticate each other. So far, several lightweight mutual authentication protocols have been claimed suitable to be employed in RFID applications [3, 7, 8, 10]. However, most of these protocols have failed to achieve the required security goals [1, 2, 4, 5, 6, 9]. Sadighian and Jalili have introduced their superior mutual authentication protocol called the Fast Lightweight Mutual Authentication Protocol (FLMAP), where designers claim their protocol guarantees tag anonymity, forward security, and location privacy whilst has resistance against ID disclosure and desynchronization attacks [10]. We, however, put FLMAP under a new rigorous test for a thorough cryptanalysis investigation for surprising results.

The rest of this paper is organized as follows: In section 2 some preliminaries and notations will be presented. In section 3 we describe FLMAP and its designing flaw. Section 4 discusses desynchronization and ID disclosure attacks against FLMAP. Section 5 proposes our solutions to fix FLMAP security flaws against investigated attacks. Concluding remarks are presented in section 6.

2. PRELIMINARIES

It is a general assumption for an RFID authentication protocol to consider the channel between the reader and the back-end database secure. Hence, in whole of this paper we do not distinguish between the reader and the back-end database of FLMAP. To describe the FLMAP, we follow the notations used by the designers of FLMAP [10] which are as follows:

- $k = k_1 \| k_2$ indicates tag's secret key.
- ID, INX indicate tag's static and dynamic identifier respectively.
- $\vee$ indicates bitwise ”OR” operation.
- $\wedge$ indicates bitwise ”AND” operation.
- $\oplus$ indicates bitwise ”XOR” operation.
- $\neg$ indicates bitwise “Not” operation.
- t_{sys} is a parameter which is used by the reader to indicate the system's time.
- t_{tag} is a parameter which is used by the tag to indicate the tag's time and updates after each successful run of protocol.
- $t_{\max}$ is the upper bound of t_{sys} which tag can accept.
- t_0 is a value which is used as the initial value of t_{tag}.
- All parameters in the protocol are of length L-bit.
- $A \to B$ refers to assigning A to B.

- $\chi \xrightarrow{\$} x$ is the experiment of uniformly choosing a random element from a finite set χ and assigning it to x.

3. FLMAP DESCRIPTION

The FLMAP, which has been depicted in Algorithm 1, runs as follows:

1. The reader sends a pseudorandom number n_1 and the system time t_{sys} to the tag.

2. The tag verifies the condition $t_{tag} < t_{sys} < t_{\max}$. If it is correct, tag generates another pseudorandom number called n_2, computes $x = k_1 \vee (n_1 \oplus n_2)$, $y = k_1 \oplus ID \oplus n_2$ and sends x, y and it's INX to the reader. Otherwise, the tag sends two meaningless pseudorandom numbers such that reader can not trace the tag and the protocol will be terminated.

3. Reader extracts n_2 from x and calculates $y' = k_1 \oplus ID \oplus n_2$ and authenticates the tag if $y' = y$. Then, the reader computes $z = k_2 \oplus (ID \wedge n_2)$ and sends it to the tag and updates keys and INX as follows:

$$k_1^{next} = k_2 \wedge (n_1 \oplus n_2)$$

$$k_2^{next} = k_1 \wedge (n_1 \oplus n_2)$$

$$INX_{next} = (INX \vee ID) \oplus n_2$$

However, if $y \neq y'$ then the protocol will be terminated.

4. The tag calculates $z' = k_2 \oplus (ID \wedge n_2)$ and verifies the condition $z = z'$. If the equality occurs, that means the reader also has been authenticated for the tag and the tag updates its t_{tag}, keys and INX as follows:

$$t_{tag} = t_{sys}$$

$$k_1 = k_2 \wedge (n_1 \oplus n_2)$$

$$k_2 = k_1 \wedge (n_1 \oplus n_2)$$

$$INX = (INX \vee ID) \oplus n_2$$

Based on the designers claim [10], to overcome the desynchronization attack the reader of FLMAP saves two records of keys and index, i.e. $(k_1^{old}, k_2^{old}, INX^{old})$ and $(k_1^{next}, k_2^{next}, INX^{next})$. However, we will present an efficient desynchronization attack against this protocol in section 4.

3.1. A FLAMP Design Issue

The designers of FLMAP claimed that after the second pass of algorithm the reader extracts n_2 from x uniquely. However, recall that $x = k_1 \vee (n_1 \oplus n_2)$, n_2 may not be uniquely extracted from x. For example, set $k_1 = 3$, $x = 3$ and $n_1 = 3$, one can see that, because of the ”OR” operation, we have four options for n_2 which are $n_2 = 0, 1, 2$ and 3. Therefore the protocol can not be run properly.

To solve the above designing flaw, one of the designers suggested [11] to replace the ”OR” operation in $x = k_1 \vee (n_1 \oplus n_2)$ with ”XOR” such that $x = k_1 \oplus (n_1 \oplus n_2)$. Hence, after modification, given x, k_1 and n_1 it is possible to extract n_2 uniquely. Hence, we present our attacks for this variant of FLMAP.

4. ATTACKS ON THE PROTOCOL

We made the following observations in the FLMAP protocol:

1. n_2 affects x, y and z through $\oplus$ operation.

2. The outputs of the operations that have been used in the key updating phase of the algorithm are biased.

3. t_{sys} has no impact on the reader calculations and it is only used by the tag as a part of reader authentication process. We use these observations to mount our attacks on FLAMP.

4.1 Desynchronization Attack

Tag and reader share different values to authenticate each other, e.g. keys and ID and update some of the shared values in each successful interaction. For example, in FLMAP, k_1, k_2, ID and INX are used through authentication process and k_1, k_2 and INX are updated after each successful interaction. However, if attacker forces tag and reader to update those values such that they can not authenticate each other any more, we say that they have been desynchronized and the attack known as a desynchronization attack. We can desynchronize the tag and the reader of FLMAP following the approach depicted in Algorithm 2. This attack is a man in the middle attack and works as follows:

1. The attacker receives the first message, i.e. (n_1, t_{sys}) and forwards it to the tag without any changes.

2. Tag evaluates $t_{tag} < t_{sys} < t_{\max}$. If the condition is correct, then tag generates n_2, computes $x = k_1 \oplus n_1 \oplus n_2$, $y = k_1 \oplus ID \oplus n_2$ and sends x, y and INX to the reader.

3. The attacker intercepts the sent x, y and INX, flips the most significant bit (MSB) of x and y, we denote the new values by x^* and y^*, passes x^*, y^* and INX to the reader.

4. The reader extracts $n_2^* = x^* \oplus k_1 \oplus n_1$ and authenticate the tag if $k_1 \oplus ID \oplus n_2^* = y^*$. Reader then

sends $z = k_2 \oplus (ID \wedge n_2^*)$ to the tag and updates k_1, k_2 and INX.

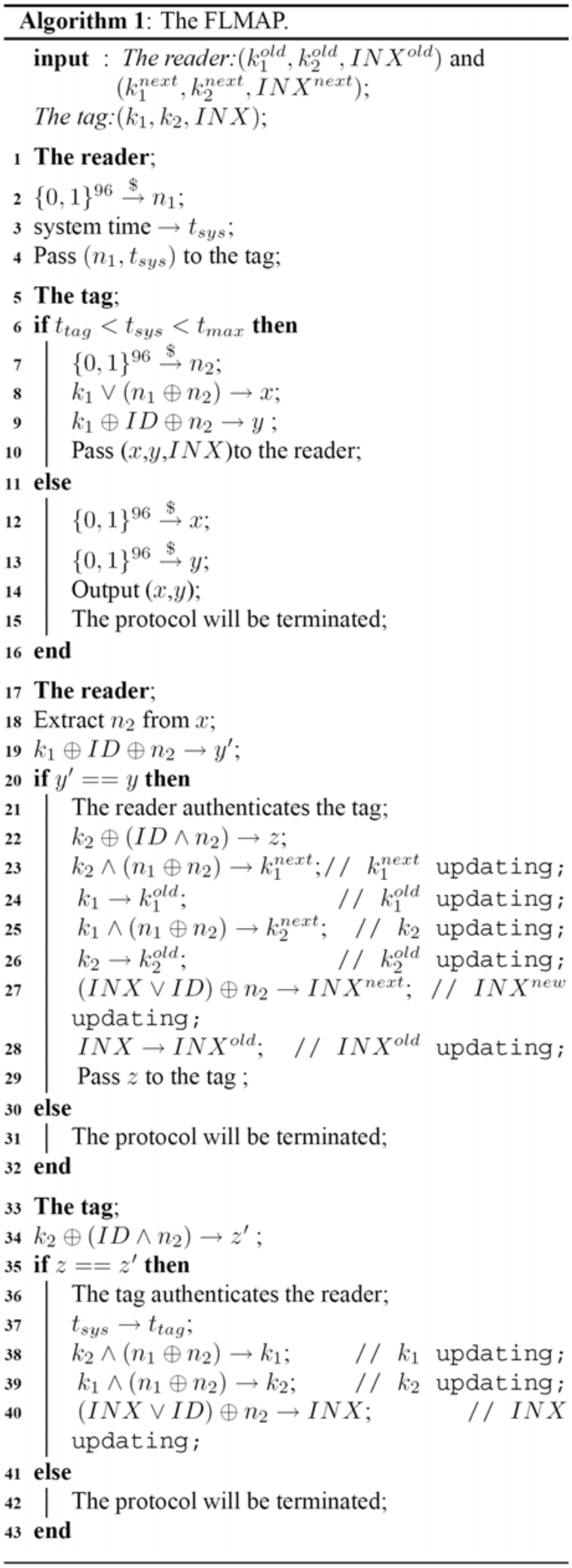
5. The attacker passes z to the tag.

6. If $k_2 \oplus (ID \wedge n_2^*) = k_2 \oplus (ID \wedge n_2)$, the tag authenticates the reader and updates k_1, k_2 and INX.

If all steps of the above attack executed, it means that the reader has used n_2^* in the updating phase while the tag has used n_2. Hence, the new values of keys and INX in the tag are not equal to those values in the reader and they can not authenticate each other any more. Now, we should determine the success probability of the above attack.

If the attacker flips the MSB of x, where the flipped value denoted by x^*, it leads to a flip in the MSB of extracted n_2 with the probability of “1”, where the extracted n_2 denoted by n_2^*. So, if we compare $k_1 \oplus ID \oplus n_2$ with $k_1 \oplus ID \oplus n_2^*$ it is obvious that they are different in MSB with the probability of “1”. On the other hand, the attacker already has flipped the MSB of y, called y^*. Hence, with the probability of “1” the reader authenticates the tag. Therefore, the success probability of this phase of attack is “1”.

Now we consider the success probability of the next phase of attack where the reader sends $z = k_2 \oplus (ID \wedge n_2^*)$ to the tag and the tag compares it with $z' = k_2 \oplus (ID \wedge n_2)$ and if $z = z'$ then authenticates the reader and updates the keys and INX. On the other hand, we know that the only different between n_2 and n_2^* is their MSB. In addition, for any given two single-bit values b and b' if one flips either b or b' the result of $b \wedge b'$ will flip with the probability of $1/2$. Hence, with the probability of



Algorithm 1: The FLMAP.

```
input : The reader:$(k_1^{old}, k_2^{old}, INX^{old})$ and
        $(k_1^{next}, k_2^{next}, INX^{next})$;
        The tag:$(k_1, k_2, INX)$;
The reader;
$\{0,1\}^{96} \xrightarrow{\$} n_1$;
system time $\rightarrow t_{sys}$;
Pass $(n_1, t_{sys})$ to the tag;
The tag;
if $t_{tag} < t_{sys} < t_{max}$ then
    $\{0,1\}^{96} \xrightarrow{\$} n_2$;
    $k_1 \vee (n_1 \oplus n_2) \rightarrow x$;
    $k_1 \oplus ID \oplus n_2 \rightarrow y$ ;
    Pass $(x, y, INX)$to the reader;
else
    $\{0,1\}^{96} \xrightarrow{\$} x$;
    $\{0,1\}^{96} \xrightarrow{\$} y$;
    Output $(x, y)$;
    The protocol will be terminated;
end
The reader;
Extract $n_2$ from $x$;
$k_1 \oplus ID \oplus n_2 \rightarrow y'$;
if $y' == y$ then
    The reader authenticates the tag;
    $k_2 \oplus (ID \wedge n_2) \rightarrow z$;
    $k_2 \wedge (n_1 \oplus n_2) \rightarrow k_1^{next}$;// $k_1^{next}$ updating;
    $k_1 \rightarrow k_1^{old}$;                      // $k_1^{old}$ updating;
    $k_1 \wedge (n_1 \oplus n_2) \rightarrow k_2^{next}$;  // $k_2$ updating;
    $k_2 \rightarrow k_2^{old}$;                      // $k_2^{old}$ updating;
    $(INX \vee ID) \oplus n_2 \rightarrow INX^{next}$;  // $INX^{new}$
    updating;
    $INX \rightarrow INX^{old}$;   // $INX^{old}$ updating;
    Pass $z$ to the tag ;
else
    The protocol will be terminated;
end
The tag;
$k_2 \oplus (ID \wedge n_2) \rightarrow z'$ ;
if $z == z'$ then
    The tag authenticates the reader;
    $t_{sys} \rightarrow t_{tag}$;
    $k_2 \wedge (n_1 \oplus n_2) \rightarrow k_1$;       // $k_1$ updating;
    $k_1 \wedge (n_1 \oplus n_2) \rightarrow k_2$;       // $k_2$ updating;
    $(INX \vee ID) \oplus n_2 \rightarrow INX$;          // $INX$
    updating;
else
    The protocol will be terminated;
end
```

$\frac{1}{2}$, the equality $z = z'$ would be true and the tag accepts the received z and does the updating phase. So, for one run of algorithm, the total success probability of attack is $\frac{1}{2}$.

***Remark* 1.** *When, at the first step of protocol, the reader sends t_{sys} to the tag, tag evaluates $t_{tag} < t_{sys} < t_{\max}$ and continues the game if this condition is true and uses t_{sys} as the t_{tag} of the next run of protocol. However, if the attacker intercepts the reader's message and replaces t_{sys} with a new t_{sys} which is extremely larger than the sent t_{sys}, but smaller than $t_{\max}$, then it may lead to desynchronization attack from the next run of protocol. To determine a reasonable upper bound for $t_{\max}$, the attacker can use a try and error process.*

4.2 ID Disclosure Attack

At the end of each interaction, the FLMAP's tag and reader update keys and INX as follows:

$$k_1^{next} = k_2 \wedge (n_1 \oplus n_2)$$
$$k_2^{next} = k_1 \wedge (n_1 \oplus n_2)$$
$$INX^{next} = (INX \vee ID) \oplus n_2$$

Since the k_1 and k_2 are 96-bit values, their average hamming weight is 48, where the hamming weight of A is defined as the number of "1" in the binary representation of A. On the other hand for any given two single-bit b and b' the result of $b \wedge b'$ would be equal to "1" if and only if $b = 1$ and $b' = 1$. Hence, for any random selection of b and b' the result of $b \wedge b'$ would be "0" with the probability of $\frac{3}{4}$. So, we expect that, after the first run of protocol, approximately 72 bits of each of k_1 and k_2 congruent to "0". After the second successful run of protocol, we expect 90 bits of each of k_1 and k_2 congruent to "0". Therefore, after several successful runs of protocol all bits of k_1 and k_2 congruent to "0". Assume $k_1 = 0$ and recall that n_1 is known for the attacker then the attacker can uniquely extract n_2 from x, where $x = k_1 \oplus (n_1 \oplus n_2)$. Now, given n_2 and y the attacker can uniquely disclose ID from y as follows, where $y = k_1 \oplus ID \oplus n_2$:

$$ID = y \oplus k_1 \oplus n_2$$

5. IMPROVING FLMAP

In spite of claimed superior feature of FLMAP, our analyses show a design flaw in the algorithm and several drawbacks on the security of this algorithm. More precisely, the proposed algorithm does not work properly for which legitimate readers and tags may fail authenticating each other, which make the protocol un acceptable. In addition, when the above flaw get fixed, we present desynchronization and ID disclosure attacks against FLMAP with negligible complexities and the success probability not less than $1/2$.

Some vulnerabilities of FLMAP that have been employed through the above attacks are as follows:

1- n_2 does not extract from x uniquely.

2- k_1^{next} and k_2^{next} in updating phase equations congruent to zero after some successful runs of protocol.

3- t_{sys} has no impact on the reader calculations and it is only used by the tag as a part of reader authentication process.

Taking the above in to account, we propose the following modifications in to the messages of mutual authentication phase and updating phase of FLMAP:

$$x = n_2 \oplus [(ID \wedge k_1) \vee (\neg ID \wedge k_2)]$$
$$y = [n_2 \wedge (ID \oplus k_1 \oplus t_{sys})] \vee [\neg n_2 \wedge (\neg ID \oplus k_2 \oplus \neg t_{sys})]$$
$$z = k_1 \oplus k_2 \oplus [(t_{sys} \wedge n_2) \vee (\neg t_{sys} \wedge ID)]$$
$$k_1^{next} = [(k_1 \wedge n_1) \vee (\neg k_1 \wedge n_2)] \oplus [(t_{sys} \wedge k_2) \vee (\neg t_{sys} \wedge ID)]$$
$$k_2^{next} = [(k_2 \wedge n_2) \vee (\neg k_2 \wedge n_1)] \oplus [(\neg t_{sys} \wedge k_1) \vee (t_{sys} \wedge ID)]$$
$$INX^{next} = [(INX \wedge (k_1 \oplus k_2)) \vee (\neg INX \wedge ID)] \oplus n_2$$

This modification leads to strength the FLMAP security against the mentioned weaknesses as follows:

1-Given *x, ID, k_1* and *k_2* then *n_2* can be extracted uniquely.

2- The k_1^{next} and k_2^{next} updating phase equations are not biased operations.

3- t_{sys} is used in all steps of algorithm. Hence, the attacker has no control over it.

6. CONCLUSION

In this work we have analyzed the designing procedure and the security features of the Fast Lightweight Mutual Authentication Protocol (FLMAP). The study reveals that FLMAP comes with some serious drawbacks in both protocol design and security of the process. We have shown that the protocol also has a flaw that could jeopardize the execution and that a desynchronization attack that desynchronizes the tag and reader in a single run with the probability of $\frac{1}{2}$ and it may disclose the ID in several runs of protocol.

All in all we do not recommend FLMAP, as it stands, to be employed in any application. However, we proposed some modifications to algorithm which improve the security of protocol against the presented attack. The security analysis of protocol against other attacks is the subject of future works.

REFERENCES

[1] M. B´ar´asz, B. Boros, P. Ligeti, K. L´oja and D. Nagy.Passive Attack Against the M2AP Mutual AuthenticationProtocol for RFID Tags. In *First International EURASIPWorkshop on RFID Technology*, Vienna, Austria, September 2007.

[2] T. Cao, E. Bertino, and H. Lei. Security analysis of the SASI protocol. *IEEE Transactions on Dependable and secure Computing*, 6(1):73–77, 2009.

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

[4] J. C. Hernandez-Castro, J. M. E. Tapiador, P. Peris-Lopez and J.-J. Quisquater. Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations. Technical Report arXiv:0811.4257, Nov 2008.

[5] T. Li and R. 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.

[6] T. Li, G. Wang, and R. H. Deng. Security Analysis on a Family of Ultra-lightweight RFID Authentication Protocols.*Journal of Software*, 3(3), March 2008.

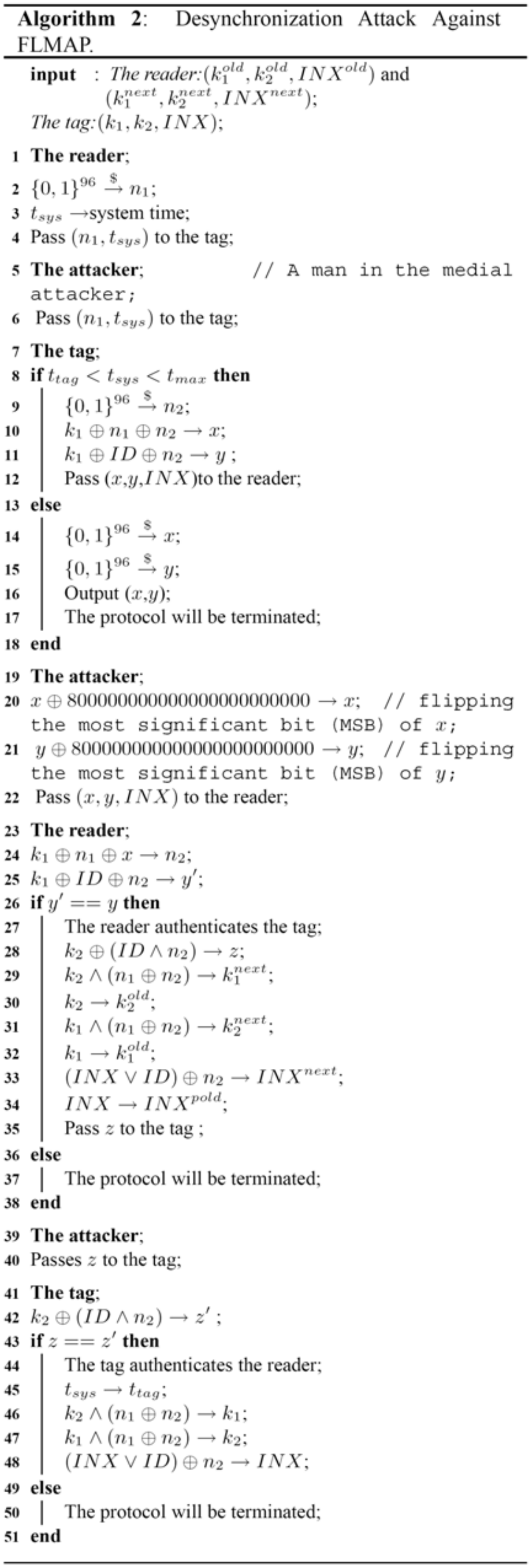
[7] P. Peris-Lopez, J. C. H. Castro, J. M. Est´evez-Tapiador, and A. Ribagorda. Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In *WISA*, pp. 56–68, 2008.

[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags. In *OTM Federated Conferences and Workshop: ISWorkshop – IS'06*, volume 4277 of *Lecture Notes in Computer Science*, pp. 352–361, Montpellier, France, November 2006.

[9] R. C.-W. Phan. Cryptanalysis of a New UltralightweightRFID Authentication Protocol - SASI. *IEEE Transactionson Dependable and Secure Computing*, 6(4):316–320, 2009.

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

[11] A. Sadighian and R. Jalili. Security and designing of FLMAP. Personal communication, May 2010.



Algorithm 2: Desynchronization Attack Against FLMAP.

```
input : The reader:$(k_1^{old}, k_2^{old}, INX^{old})$ and
        $(k_1^{next}, k_2^{next}, INX^{next})$;
  The tag:$(k_1, k_2, INX)$;
The reader;
$\{0,1\}^{96} \xrightarrow{\$} n_1$;
$t_{sys}$ →system time;
Pass $(n_1, t_{sys})$ to the tag;
The attacker;            // A man in the medial attacker;
Pass $(n_1, t_{sys})$ to the tag;
The tag;
if $t_{tag} < t_{sys} < t_{max}$ then
    $\{0,1\}^{96} \xrightarrow{\$} n_2$;
    $k_1 \oplus n_1 \oplus n_2 \rightarrow x$;
    $k_1 \oplus ID \oplus n_2 \rightarrow y$ ;
    Pass $(x$,$y$,$INX)$to the reader;
else
    $\{0,1\}^{96} \xrightarrow{\$} x$;
    $\{0,1\}^{96} \xrightarrow{\$} y$;
    Output $(x$,$y)$;
    The protocol will be terminated;
end
The attacker;
$x \oplus 800000000000000000000000 \rightarrow x$;  // flipping the most significant bit (MSB) of $x$;
$y \oplus 800000000000000000000000 \rightarrow y$;  // flipping the most significant bit (MSB) of $y$;
Pass $(x, y, INX)$ to the reader;
The reader;
$k_1 \oplus n_1 \oplus x \rightarrow n_2$;
$k_1 \oplus ID \oplus n_2 \rightarrow y'$;
if $y' == y$ then
    The reader authenticates the tag;
    $k_2 \oplus (ID \wedge n_2) \rightarrow z$;
    $k_2 \wedge (n_1 \oplus n_2) \rightarrow k_1^{next}$;
    $k_2 \rightarrow k_2^{old}$;
    $k_1 \wedge (n_1 \oplus n_2) \rightarrow k_2^{next}$;
    $k_1 \rightarrow k_1^{old}$;
    $(INX \vee ID) \oplus n_2 \rightarrow INX^{next}$;
    $INX \rightarrow INX^{pold}$;
    Pass $z$ to the tag ;
else
    The protocol will be terminated;
end
The attacker;
Passes $z$ to the tag;
The tag;
$k_2 \oplus (ID \wedge n_2) \rightarrow z'$ ;
if $z == z'$ then
    The tag authenticates the reader;
    $t_{sys} \rightarrow t_{tag}$;
    $k_2 \wedge (n_1 \oplus n_2) \rightarrow k_1$;
    $k_1 \wedge (n_1 \oplus n_2) \rightarrow k_2$;
    $(INX \vee ID) \oplus n_2 \rightarrow INX$;
else
    The protocol will be terminated;
end
```