On Designing Lightweight RFID Security Protocols for Medical IoT

Masoumeh Safkhani1, Ygal Bendavid2, Samad Rostampour2,3, Nasour Bagheri4

1 Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran, Postal code: 16788-15811, Tel/fax:+98-21-22970117, Safkhani@srttu.edu
2 Université du Québec à Montréal, Montreal, Canada, bendavid.ygal@uqam.ca
3 Islamic Azad University, Ahvaz, Iran, samad.rostampour@iauahvaz.ac.ir
4 Electrical Engineering Department, Shahid Rajaee Teacher Training University, Iran, NBagheri@srttu.edu

Abstract. Recently, in IEEE Transactions on Industrial Informatics, Fan et al. proposed a lightweight RFID protocol which has been suggested to be employed for protecting the Medical Privacy in an IoT system. However, the protocol has trivial flaws, as it is shown recently by Aghili et al., in Future Generation Computer Systems. Aghili et al. also proposed an improved version of the protocol, based on the similar designing paradigm, called SecLAP. Although the protocol’s designers claimed full security against all attacks, we show that the proposed protocol has serious security flaws, by presenting traceability and passive secret disclosure attacks against this protocol. More precisely, we present passive partial secret disclosure attack with the complexity of eavesdropping one session of the protocol and success probability of ‘1’. The disclosed parameters can be used to trace the tag/reader in any later session which compromises the tag/reader privacy. In addition, we present a passive full secret disclosure attack against SecLAP which can disclose $2^n$-bit secret key, $n$-bit TID and $n$-bit RID with the computational complexity of $27n^7$. In addition, we show that, as it is expected, Fan et al.’s protocol has security flaws in random oracle model, where the adversary’s advantage after $q$ queries to distinguish the protocol from a random oracle is $1 - 2^{-q}$. We also evaluate the security of SecLAP in the random oracle model and show that it is as insecure as its predecessor.

Keywords: RFID, Authentication, Ultralightweight, SecLAP, Passive Attack, Random Oracle Model

1 Introduction

At any stage in the medication process, a medication error could occur. A medication error can influence appropriate prescribing, order communication, product labeling, packaging, compounding, dispensing, distribution or administration.
To reduce errors throughout a medication process, a wide variety of information technologies have been applied so far, e.g. computerized prescriber order entry, robotics, automated dispensing devices such as Near Field Communication (NFC) and Radio Frequency Identification (RFID) and the electronic medication administration record such as employing Implantable Medical Devices (IMD). Although employing information technologies in a medication system can reduce various errors and improve medication process’s quality but note it has its own concerns, especially regarding patient privacy. Camara et al. in [10] presented a comprehensive survey on the security and privacy issues in IMD. Among the recent researches, in [24], Wu et al. presented a survey of access control schemes for IMD, with a focus on the security incidents, IMD threat model and the development of regulations for IMD security. In [23], Wazid et al. proposed NFC based authentication protocol for medicine anti-counterfeiting system in the Internet of Things (IoT) environment to check the authenticity of pharmaceutical products.

RFID is a promising wireless technology which can be employed to identify or track an object in many applications and it is a vital part of an IoT based applications, given that RFID tags are very cheap and can be attached to any object to gather the required information. In an IoT based medication system, RFID tags can be attached to objects to monitor appropriate medication process or be implanted in a patient’s body to monitor the critical signals or keep a history of the received medication services, which provide health care practitioners advantages to enhance clinical practice [27]. However, those tags are very constrained and may not be able to support a standard security protocol. Hence, the privacy of tag holders, e.g. a patient, could be the main concern in RFID based applications. To address this concern, several solutions/criticisms have been proposed in the literature, e.g. [18,25,20]. A survey of these protocols is accessible through [9].

Fan et al. [14] recently proposed a lightweight RFID mutual authentication scheme and suggested to be employed for medical privacy protection in IoT. They have analyzed the security of the proposed protocol and claimed that it provides tag anonymity, replay attack resistance, synchronization attack resistance, forward secrecy, mutual authentication, and DoS attack resistance. However, it comes out very soon that the proposed protocol suffers from trivial weaknesses, as it is shown by Aghili and Malla [1] and more comprehensively later by Aghili et al. [2]. In those works, the security of the proposed scheme is scrutinized and important security pitfalls are shown. More precisely, they present an efficient secret disclosure attack and traceability attacks that violate the designers’ claims. In addition, Aghili et al. [2] also proposed an improved protocol called SecLAP following the same designing strategy and claimed optimum security against an active adversary who can control the channel between the tag and the reader and also the reader and the server.

In this paper, we evaluate the security of SecLAP [2], as the last member of this stream. Our analysis demonstrates that their protocol is as insecure as its predecessor and a passive adversary can disclose secret parameters that are
shared between the protocol parties which is enough to mount other attacks such as traceability or desynchronization attack. Although they considered the channel between the reader and the server insecure also, we adopt the proposed attack for the case where that channel is secure and the adversary only has access to the channel between the reader and the tag and not the channel between the reader and the server.

The Paper’s Contribution: This paper’s contribution has the bellow folds:

1. We investigate the security of Fan et al.’s protocol in random oracle model (ROM) and show that the adversary’s advantage to distinguish it from an ideal protocol, in which the adversary should not be able to link any two messages transferred over insecure channel together, is $1 - q^{-2}$, where $q$ the complexity of eavesdropped messages. Although Fan et al.’s protocol is already known to be insecure, but this is another look at the protocol which provide better understanding the design.

2. We provide the first third party security analysis of SecLAP, which is the successor of Fan et al.’s protocol and has been very recently published in “Future Generation Computer Systems”. SecLAP could be the latest attempt to design a secure ultra lightweight protocol for constrained environments. However, our analysis demonstrated that it is not much secure compared to Fan et al.’s protocol. All attack presented against this protocol are in passive adversary model. More precisely we present a passive attack which can disclose partially secret parameters of the protocol with the complexity of eavesdropping of one session of the protocol and negligible computational complexity; it can be done even by pen and paper. The extracted information are enough to trace the tag or the reader in later sessions of the protocol. In addition we present a full secret disclosure attack that can extract all secret parameters of the protocol with the complexity of $27n^7$.

3. In the discussion part of the paper, we argue that it may be better to do not attempt to design a secure ultra-lightweight protocol, because it could be simply “Impossible Mission”.

Paper Organization: The required preliminaries and the notations that are used in this paper along with a brief description of Fan et al. [14] and SecLAP protocols are presented in Section 2. A review of the security of Fan et al.’s protocol is presented in Section 3. We present the result of our investigation on the security level of SecLAP protocol in Section 4. We discuss our suggestions to improve the protocol and conclude the paper in Section 5.

2 Preliminaries

2.1 Notations

Table 1 depicts the notations used in the rest of this paper. Fan et al. introduced an special bit-oriented operation in their protocol, called $\text{Cro}(X, Y)$. Given $X = X_0, ..., X_{n-1}$ and $Y = Y_0, ..., Y_{n-1}$ as the binary representation of $X$ and
respectively, where \( X_i \) denotes the \( i^{th} \) bit of string \( X \), the \( \text{Cro}(X,Y) = O_0, ..., O_{2n-1} \) operation is defined as below, for \( i = 0, ..., [n - 1] \):

\[
\begin{align*}
    O_{2i} &= (\bar{X} \parallel Y)_{2i} \oplus (\bar{Y} \parallel X)_{2i+1} \\
    O_{2i+1} &= (\bar{X} \parallel Y)_{2i+1} \oplus (\bar{Y} \parallel X)_{2i}
\end{align*}
\]

We provide an example to show the functionality of \( \text{Cro}(X,Y) \) as follows:

\[
\begin{align*}
    X &= 0110\ 0101; \\
    Y &= 1011\ 0111; \\
    \bar{X} &= 1001\ 1010; \\
    \bar{Y} &= 0100\ 1000; \\
    \bar{Y} \parallel X &= 0100\ 1000\ 0110\ 0101; \\
    \bar{X} \parallel Y &= 1001\ 1010\ 1011\ 0111; \\
    \text{Cro}(X,Y) &= 0001\ 1110\ 0010\ 1101;
\end{align*}
\]

Based on this expression, the length of \( \text{Cro}(X,Y) \) will be equal to the length of \( \bar{X} \parallel Y \) and \( \bar{Y} \parallel X \) which is twice that of \( X \) or \( Y \). On the other hand, later we see that this function is used to update a secret value \( K \) as \( K_{\text{new}} = \text{Cro}(N_R \oplus N_S \oplus N_T, K) \) which makes confusion, as the length of \( K_{\text{new}} \) then will be twice the length of \( K \). However, following personal communication with the protocol designers [13], they stated that to reduce the length of the produced \( K_{\text{new}} \), XOR operation is performed on the top and the bottom of the \( \text{Cro}() \) operation result, see Fig. 1. In this way, the length of \( K_{\text{new}} \) and \( K_{\text{old}} \) would be the same, before the next round of authentication begins.

\[
\begin{align*}
    X &= 0110\ 0101; \\
    Y &= 1011\ 0111; \\
    \bar{X} &= 1001\ 1010; \\
    \bar{Y} &= 0100\ 1000; \\
    \bar{Y} \parallel X &= 0100\ 1000\ 0110\ 0101; \\
    \bar{X} \parallel Y &= 1001\ 1010\ 1011\ 0111; \\
    \text{Cro}(X,Y) &= 0001\ 1110\ 0010\ 1101;
\end{align*}
\]

Fig. 1. To reduce the output length of \( \text{Cro}(X,Y) \) [13]

To improve the Fan et al.’s protocol, Aghili et al. proposed a new component called modular rotation function \( \text{MRot}_{(K)}(X,Y) \) which is used in the structure of \( \text{SecLAP} \). Given two \( n \)-bit strings \( X \) and \( Y \) and a \( 2n \)-bit string \( K = k_1 \parallel k_0 \), the modular rotation function \( \text{MRot}_{(K)}(X,Y) \) is defined as follows:
Table 1. Notations used in this paper

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>An RFID reader</td>
</tr>
<tr>
<td>RID</td>
<td>The identification value of the reader R</td>
</tr>
<tr>
<td>S</td>
<td>A cloud server</td>
</tr>
<tr>
<td>T</td>
<td>An RFID tag</td>
</tr>
<tr>
<td>TID</td>
<td>The identification (ID) of the tag T</td>
</tr>
<tr>
<td>K</td>
<td>The current session number</td>
</tr>
<tr>
<td>$K_{new}$</td>
<td>The new session number</td>
</tr>
<tr>
<td>PRNG()</td>
<td>The pseudo random number generator</td>
</tr>
<tr>
<td>Cro(x, y)</td>
<td>The bit-oriented operation defined by Fan et al.</td>
</tr>
<tr>
<td>Rot(x, y)</td>
<td>The rotation of string x based on the hamming weight of y, i.e. $W(y)$</td>
</tr>
<tr>
<td>Rot(x, y)</td>
<td>The rotation of string x to left</td>
</tr>
<tr>
<td>MRot($K$, x, y)</td>
<td>The bit-oriented operation defined which is used in SecLAP</td>
</tr>
<tr>
<td>od(x)</td>
<td>Odd bits of string x.</td>
</tr>
<tr>
<td>ev(x)</td>
<td>Even bits of string x.</td>
</tr>
<tr>
<td>⊕</td>
<td>The bitwise XOR operation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark</td>
<td>Two temporary bits that are used to indicate the status of the last session</td>
</tr>
<tr>
<td>X</td>
<td>The bitwise complement of the string X</td>
</tr>
</tbody>
</table>
– The odd bits of $X$ are concatenated with the even bits of $Y$, the result is XORed by the bits of $k_1$ and the result is rotated to the left depending on the value of $(Y \oplus k_1) \mod n$. The result is used as the odd bits of the final result. We can represent it as $[(od(X)||ev(Y)) \oplus k_1] \ll [(Y \oplus k_0) \mod n]$, where $od(X)$ and $ev(Y)$ respectively denote the odd bits of $X$ and the even bits of $Y$.

– The even bits of $X$ are concatenated with the odd bits of $Y$, the result is XORed by the bits of $k_0$ and the result is rotated to the left depending on the value of $(Y \oplus k_2) \mod n$. The result is used as the even bits of the final result. We can represent it as $[(ev(X)||od(Y)) \oplus k_0] \ll [(Y \oplus k_1) \mod n]$.

It is clear $MRot_{(K)}(X,Y)$ is more complicated compared to $Cro(X,Y)$ and it is also a function of the secret key $K$. Thanks to this property, the designers expected to provide optimum security for their protocol when they use $MRot_{(K)}(X,Y)$ as the core function and the source of confusion. It should be noted the output length of $MRot_{(K)}(X,Y)$ function, similar to $Cro(X,Y)$, will be equal to twice that of $X$ or $Y$.

### 2.2 Random Oracle Model

A Random Oracle \cite{8} is a theoretical black box which responds to every query with a (truly) random response chosen uniformly from its output domain. Its output to the identical inputs is the same. A Random Oracle, denoted by $R$, is defined by $R : \{0,1\}^* \to \{0,1\}^\infty$. Here $R$ is a function chosen uniformly at random from the set of all functions with the same domain and range.

In general (as in our paper), the output’s length of $R$ is predefined to some fixed value, e.g. $b$ bits. Hence, we re-define $R$ as $R_b : \{0,1\}^* \to \{0,1\}^b$. Given an input $x \in \{0,1\}^*$, $R_b$ will give $y = R_b(x) \in \{0,1\}^b$ as the output.

**Adversary:** We consider a computationally unbounded adversary with access to $R_m$ or a real cryptographic component $C$, which could be a cryptographic protocol $P$ and $m$ is the total length of the observable messages in each query. The adversary’s “running time” is determined by the number of oracle queries that it makes to $R_m/C$. We use the symbol $O$ (big-Oh), for “the expected running time of at most” and $\Omega$ (big-Omega), for “the expected running time not less than”.

**Indistinguishability** In the cryptology terminology, $R_m$ is considered as an ideal system and the distinguisher tries to distinguish the candidate crypto system $C$ with $m$-bit output length from $R_m$. In the framework of indistinguishability, the distinguisher faced with either $C$ or $R_m$ and aims to understand whether interacts with $C$ or $R_m$. Now we present the formal definition of indistinguishability following \cite{16}:
Definition 1. \( C \) and \( R_m \) are (computationally) indistinguishable if for any (computationally efficient) algorithm \( D \) (called distinguisher), interacting with one of these components and generating a binary output (0 or 1), it holds that:

\[
|Pr[D^C = 1] - Pr[D^{R_m} = 1]| < \epsilon
\]

where \( \epsilon \) is a negligible function of the security parameter \( k \).

In this framework, the maximum number of queries is bounded and denoted by \( q \).

Definition 2. A crypto system \( C \) is said to be \((t_D, q, \epsilon)\) indistinguishable from an ideal primitive \( R_m \) if for any distinguisher \( D \) it holds that:

\[
|Pr[D^C = 1] - Pr[D^{R_m} = 1]| < \epsilon
\]

The distinguisher runs in time at most \( t_D \) and makes at most \( q \) queries to \( C / R_m \). \( C \) is said to be (computationally) indistinguishable from \( R_m \) if \( \epsilon \) is a negligible function of the security parameter \( k \).

### 2.3 Fan et al.’s Protocol Description

The Fan et al. authentication protocol, as depicted in Fig. 2, proceeds as below:

1. The reader starts the authentication by generating a new random number \( N_R \) and sending it along with Query to the tag.
2. Once the tag receives the message, it:
   - generates a random number \( N_T \);
   - computes \( \text{Cro}(\text{RID} \oplus \text{TID}, K) \);
   - sets \( \text{Mark} = 00 \)
   - and sends \( N_T \) and \( \text{Cro}(\text{RID} \oplus \text{TID}, K) \) to the reader.
3. Upon receipt of the message, the reader stores \( N_T \) and sends \( N_R, N_T \) and \( \text{Cro}(\text{RID} \oplus \text{TID}, K) \) to the server.
4. When receives the message, the server:
   - stores \( N_R \) and \( N_T \);
   - searches its database for any entry related to received \( \text{Cro}(\text{RID} \oplus \text{TID}, K) \);
   - generates another random number \( N_S \);
   - computes \( \text{Cro}(\text{RID} \oplus \text{TID}, K \oplus N_S), \text{Rot}(K \oplus \text{TID}, K \oplus \text{RID}) || (N_S \oplus K) \) and sends them to the reader.
5. Once the reader receives the message, it:
   - retrieves \( \text{TID} \) and \( N_S \) from \( \text{Rot}(K \oplus \text{TID}, K \oplus \text{RID}) \) and \( N_S \oplus K \) respectively;
   - computes \( \text{Cro}(\text{RID} \oplus \text{TID}, K \oplus N_S) \) and checks whether it equals with the received \( \text{Cro}(\text{RID} \oplus \text{TID}, K \oplus N_S) \); if it does not then it stops the protocol otherwise calculates \( \text{TID} \oplus N_R \);
   - and sends \( \text{TID} \oplus N_R \) and \( N_S \) to the tag.
6. Once receipt of the message, the tag:
   - retrieves \( \text{TID} \) as \( \text{TID} \oplus N_R \oplus N_R \);
checks whether retrieved $TID$ equals with its $TID$, if it does not then it stops the protocol otherwise updates its key as $K_{new} = Cro(N_R \oplus N_S \oplus N_T, K)$;

- computes $Cro(RID \oplus TID, K_{new})$;
- and sends $Cro(RID \oplus TID, K_{new})$ to the reader.

7. When the reader receives the message, it:
- calculates $Cro(RID \oplus TID, Cro(N_R \oplus N_S \oplus N_T, K))$ and checks whether it equals to the corresponding received value, if it does not then it terminates the protocol otherwise updates $K$ as $K_{new} = Cro(N_R \oplus N_S \oplus N_T, K)$
- and sends $Cro(RID \oplus TID, K_{new})$ to the server.

8. Once the server receives the message, it:
- calculates $Cro(RID \oplus TID, Cro(N_R \oplus N_S \oplus N_T, K))$ and checks whether it equals to the corresponding received value, if it does not then it terminates the protocol otherwise updates $K$ as $K_{new} = Cro(N_R \oplus N_S \oplus N_T, K)$;
- and sends $K_{new} \oplus N_T \oplus N_R$ to the reader.

9. Upon receipt of the message, the reader extracts $K_{new}$, verifies it and if it passed the verification, sends $K_{new} \oplus N_T \oplus N_R$ to the tag.

10. The tag verifies the correctness of $K_{new}$ by computing $K_{new} \oplus N_T \oplus N_R \oplus N_T \oplus N_R$ and if $K_{new}$ passed the verification, tag sets $Mark = 01$ and sends $Mark \oplus N_S$ to the server through the reader.

11. The server:
- checks whether $Mark$ equals to 01, if it is, a new record $\{Cro(RID \oplus TID, K_{new}), Rot(K_{new} \oplus TID, K_{new} \oplus RID)\}$ will be generated and added in its database;
- sends a record completion notification to the tag via the reader.

12. When the tag received the message, sets $Mark = 10$.

2.4 Description of SecLAP

SecLAP has been proposed to improve the security drawbacks of its predecessor protocol by Fan et al. based on a similar designing paradigm, proceeds as below and also is depicted in Fig. 3:

1. The reader starts the authentication by generating a new random number $N_R$ and sending it along with $Query$ to the tag.

2. Once the tag receives the message, it:
   - generates a random number $N_T$;
   - computes $M_1 = MRot(K_i)(TID \oplus N_R, od(K_i) \oplus N_T)$ and $M_2 = MRot(K_i)(ev(K_i) \oplus N_R, TID \oplus N_T)$;
   - and sends $N_T || M_1 || M_2$ to the reader;
Fig. 2. Mutual authentication phase of Fan et al.'s protocol [14]
3. Upon receipt of the message, the reader obtains $TID$ from $M_1$ and verifies the received $M_2$ based on it. Then, the reader computes $IDX_i = MRot(K_i)(RID \oplus od(K_i), TID)$, $IDC_i = MRot(K_i)(ev(K_i) \oplus TID, RID)$, $M_3 = IDX_i \oplus K_0$ and $M_4 = MRot(K_i)(NT \oplus od(IDC_i), ev(IDC_i) \oplus NR)$, and forwards $NT||NR||M_3||M_4$ to the server. If the tag has not been authenticated, the protocol is terminated.

4. When receives the message, the server:
   - extracts $IDX_i$ and finds related $K_i$ and $IDC_i$ in its database $IDT$ and verifies the received $M_4$;
   - generates another random number $N_S$;
   - computes $M_5 = MRot(K_i)(ev(K_i) \oplus N_S, od(IDC_i) \oplus N_R)$, $M_6 = MRot(K_i)(od(K_i) \oplus N_S, ev(IDC_i) \oplus N_S)$ and sends $M_5||M_6$ to the reader.

5. Once the reader receives $M_5||M_6$, it:
   - retrieves $N_S$ from $M_6$ and verifies the received $M_5$ accordingly;
   - computes $M_7 = MRot(K_i)(od(K_i) \oplus N_S, TID \oplus N_T)$ and $M_8 = MRot(K_i)(ev(K_i) \oplus N_S, TID \oplus N_S)$;
   - sends $M_7||M_8$ to the tag.

6. Once receipt the message $M_7||M_8$, the tag:
   - retrieves $N_T$ from $M_7$ and verifies $M_8$ to authenticate the server/reader;
   - calculates $K_{i+1} = MRot(K_i)(od(K_i) \oplus N_S \oplus N_R \oplus N_T \oplus ev(K_i), TID)$;
   - computes $M_9 = MRot(K_i)(od(K_{i+1}) \oplus TID, N_R \oplus N_S)$ and $M_{10} = MRot(K_i)(od(K_{i+1}) \oplus TID \oplus ev(K_{i+1}), N_S)$;
   - sends $M_9||M_{10}$ to the reader.

7. When the reader receives the message, it:
   - calculates $K_{i+1} = MRot(K_i)(od(K_i) \oplus N_S \oplus N_R \oplus N_T \oplus ev(K_i), TID)$ and verifies the received $M_9||M_{10}$;
   - computes $M_{11} = MRot(K_i)(ev(IDC_i) \oplus ev(K_{i+1}), N_R \oplus N_S)$, $M_{12} = MRot(K_i)(od(IDC_i) \oplus ev(K_{i+1}), N_R \oplus N_S), IDX_{i+1} = MRot(K_i)(RID \oplus od(K_{i+1}), TID)$, $IDC_{i+1} = MRot(K_i)(TID \oplus ev(K_{i+1}), RID)$, $M_{13} = IDX_{i+1} \oplus K_i$, $M_{14} = MRot(K_i)(ev(IDC_{i+1}) \oplus M_{11}, N_R \oplus N_S)$, $M_{16} = MRot(K_i)(od(IDC_{i+1}) \oplus ev(IDC_{i+1}), N_S)$;
   - sends $M_{11}||M_{12}||M_{13}||M_{14}||M_{15}||M_{16}$ to the server.

8. Once the server receives the message, it:
   - extracts $K_{i+1}$, $IDX_{i+1}$ and $IDC_{i+1}$ respectively from $M_{11}$, $M_{12}$, $M_{13}$, $M_{14}$ and $M_{15}$ and verifies $M_{16}$ accordingly to update $K_i$ and $K_{i+1}$;
   - adds $(IDX_{i+1}, K_{i+1}, IDC_{i+1})$ to its database $IDT$;
   - computes $M_{17} = MRot(K_i)(od(K_{i+1}) \oplus ev(IDC_{i+1}), N_S \oplus N_R)$ and sends it to the reader.

9. Upon receipt of the message $M_{17}$, the reader verifies it to decide whether update $K_i$ as $K_{i+1}$. If $M_{17}$ has been verified correctly, the reader computes $M_{18} = MRot(K_i)(ev(K_{i+1}) \oplus NT \oplus od(K_{i+1}), TID \oplus N_S)$ and sends it to the tag.

10. The tag verifies the correctness of received $M_{18}$ to update $K_i$ as $K_{i+1} = MRot(K_i)(od(K_i) \oplus N_S \oplus NR \oplus NT \oplus ev(K_i), TID)$ and the protocol is completed.
<table>
<thead>
<tr>
<th>Server $S(M\text{Rot}(K_i)),$</th>
<th>Reader $R(\text{RID}, M\text{Rot}(K_i)),$</th>
<th>Tag $T(\text{TID}, M\text{Rot}(K_i)),$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Generates $N_R$</td>
<td>(2) $\text{ver}_{NS}, M_1 \leftarrow M\text{Rot}(K_i)(\text{TID} \oplus \text{NT}) \oplus N_R \oplus N_T$</td>
<td>(3) Generates $N_T,$ $M_1 \leftarrow M\text{Rot}(K_i)(\text{TID} \oplus \text{NT}) \oplus N_R \oplus N_T$</td>
</tr>
<tr>
<td>(4) $\text{M}_2 \leftarrow M\text{Rot}(K_i)(\text{ev}(K_i)) \oplus \text{NT}, \text{TID} \oplus N_T$</td>
<td></td>
<td>(4) $\text{M}_2 \leftarrow M\text{Rot}(K_i)(\text{ev}(K_i)) \oplus \text{NT}, \text{TID} \oplus N_T$</td>
</tr>
<tr>
<td>(5) Retrieves $\text{TID}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I\text{DX}_{i+1} = M\text{Rot}(K_i)(\text{RID} \oplus \text{ev}(K_i)) \oplus \text{NT}, \text{TID} \oplus N_T \oplus N_R$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6) $\text{M}_3 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7) Retrieves $I\text{DX}_{i+1},$ searches in its database to find $K_i$ and $M_i \leftarrow M\text{Rot}(K_i)$ and $N_g = \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G$</td>
<td>(8) $\text{M}_5 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G$</td>
<td>(9) Retrieves $N_g$ from $M_5,$ $\text{M}_6 \leftarrow M\text{Rot}(K_i) \oplus \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G$</td>
</tr>
<tr>
<td>(10) $\text{M}<em>7 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(I\text{DC}</em>{i+1}), N_g \oplus N_R$</td>
<td>(11) Retrieves $N_g$ from $M_7$ and verifies $\text{M}<em>8 \leftarrow M\text{Rot}(K_i)(\text{ev}(K_i)) \oplus \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1})$</td>
<td></td>
</tr>
<tr>
<td>(12) $\text{M}<em>9 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td>(13) Calculate $\text{K}_{i+1}$ and verifies the received $\text{M}<em>7 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td></td>
</tr>
<tr>
<td>(14) $\text{M}<em>1 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td>(15) extracts $K_{i+1}, I\text{DX}_{i+1}$ from $\text{M}<em>1 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td></td>
</tr>
<tr>
<td>(16) $\text{M}<em>7 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus N_G \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td></td>
<td>(17) Checks correctness of $\text{M}<em>1 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
</tr>
<tr>
<td>(18) $\text{M}<em>8 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td>(19) Checks correctness of $\text{M}<em>1 \leftarrow \text{NT}, \text{TID} \oplus N_T \oplus N_R \oplus \text{ev}(\text{K}</em>{i+1}), N_g$</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Mutual Authentication phase of SecLAP
3 Security Analysis of Fan et al.’s Protocol

3.1 Secret Disclosure and Traceability Attacks

Fan et al.’s protocol has trivial vulnerabilities, also reported by Aghili and Mala [1]. More precisely, consider a passive adversary which just eavesdrops the transferred messages between a legitimate reader $R$ and the target tag $T$. In this case, the adversary achieves the following set of information that is transferred over an insecure channel between $T$ and $R$:

- $N_R$ sent from $R$ to $T$;
- $Cro(RID \oplus TID, K)$ and $N_T$ sent from $T$ to $R$;
- $TID \oplus N_R$ and $N_S$ sent from $R$ to $T$;
- $Cro(RID \oplus TID, K_{new})$ sent from $T$ to $R$;
- $K_{new} \oplus N_T \oplus N_R$ sent from $R$ to $T$;
- $Mark \oplus N_S$ sent from $T$ to $R$;

Given the above information, it would be trivial to extract, $TID$ and $K_{new}$, as follows:

- $TID = (TID \oplus N_R) \oplus N_R$;
- $K_{new} = (K_{new} \oplus N_T \oplus N_R) \oplus N_T \oplus N_R$.

Given that $TID$ is a constant parameter which determines the tag identity, it can be employed to trace the tag holder. This fact compromises the security of the protocol against traceability and tag anonymity.

Remark 1. This attack does not depend on the definition of $Cro(X, Y)$ and works for any $Cro(X, Y)$. Hence it is not possible to improve the security of the protocol against this attack by just using a more complicated function instead of the current $Cro(X, Y)$.

On the other hand, on Step 6, the tag sends $Cro(RID \oplus TID, K_{new})$ to the reader over public channel. Now assume that the adversary also eavesdrops that message between $R$ and $T$, where the values of $TID$ and $K_{new}$ have been extracted by the adversary already.

Given $Cro(RID \oplus TID, K')$, we denote $C_i = (Cro(RID \oplus TID, K'))_i$ for simplicity, i.e. its $i^{th}$ bit. Given that $\mathcal{X} = \mathcal{X} \oplus 1$ any $\mathcal{X} \in \{0, 1\}$, for an even value of $n$, we can extract the following equations, for different cases of $i$:

$$
\begin{cases}
0 \leq i < \frac{n}{2} - 1 & C_{2i} = RID_{2i+1} \oplus TID_{2i+1} \oplus K'_{2i} \\
C_{2i+1} = RID_{2i} \oplus TID_{2i} \oplus K'_{2i+1} \\
\frac{n}{2} \leq i \leq n-1 & C_{2i+1} = RID_{2i+1} \oplus TID_{2i+1} \oplus 1 \oplus K'_{2i+1} \\
C_{2i+1} = RID_{2i} \oplus TID_{2i} \oplus 1 \oplus K'_{2i+1} \oplus 1
\end{cases}
$$

(1)

where $n$ is the length of parameters in the protocol, e.g. $TID \in \{0, 1\}^n$. In Eq. [1] exclude $RID_j$ for $0 \leq j < n - 1$, all variables are known. Hence, $RID$ will be retrieved easily from that linear equations. Given $RID$, the reader anonymity is compromised.
Remark 2. It should be noted the success probability of the presented attack to recover RID depends on the exact definition of Cro(X, Y). Hence, it may be possible to improve the security of the protocol against this attack by just using a more complicated function instead of the current Cro(X, Y).

3.2 Fan et al.’s Protocol Distinguishability in ROM

Any cryptographic scheme is expected to behave unpredictable form the adversary’s point of view. To model this property, indistinguishability from a random oracle is used, which we defined in Section 2.2. In this section, we show that Fan et al.’s protocol is not a secure protocol in ROM. Given that for any value of x we can state that $x \oplus 1 = x \oplus 0 = x$, Eq. [1] can be rewritten as follows:

$$
\begin{align*}
0 \leq i & \leq \frac{n}{2} - 1 \quad C_{2i} = RID_{2i+1} \oplus TID_{2i+1} \oplus K_{2i}^r \\
\frac{n}{2} \leq i & \leq n - 1 \quad C_{2i+1} = RID_{2i} \oplus TID_{2i} \oplus K_{2i+1}^r \\
C_{2i+1} & = RID_{2i} \oplus TID_{2i} \oplus K_{2i+1}^r
\end{align*}
$$

(2)

It is clear that, for $i \in \{0, 2, \ldots, n - 2\}$ and for even values of $n$, $C_i = C_{n+i+1}$ and $C_{i+1} = C_{n+i}$. This property can be used as a metric to determine that the used protocol is the Fan et al.’s protocol and not $R$. Moreover, Fan et al.’s protocol has zero-sum property. More precisely, it is easy to show that:

$$\oplus_{i=0}^{2n-1} C_i = C_0 \oplus C_1 \oplus \ldots \oplus C_{2n-1} = 0.$$

(3)

It comes for the fact that any variable appears in the above summation exactly two times, e.g. $RID_0$ or $K_0$, and for any $X \in \{0, 1\}$ and any value of $x$ we have $x \oplus X \oplus X = x \oplus 0 = x$. This property can be used as a metric to determine that the used protocol is the Fan et al.’s protocol or $R$.

To distinguish Fan et al.’s protocol $P$ from $R_m$, the adversary does as follows:

1. eavesdrops a message produced by Cro(.) function, transferred over channel, e.g. $C = Cro(K \oplus TID, K)$.
2. if $\oplus_{i=0}^{2n-1} C_i = 0$ returns $P$; otherwise returns $R_m$.

To determine the adversary’s advantage, it is clear if the adversary communicates with Fan et al.’s protocol then with the probability of ‘1’ returns $P$ in Step 2 while if it communicates with $R_m$ then returns $P$ with the probability of ‘2’. To improve the adversary’s advantage, we can increase the number of eavesdropped messages. For an adversary who eavesdropped $q$ messages, all produced by Cro(.) function, the advantage will be as follows:

$$|Pr[D^C = 1] - Pr[D^{R_m} = 1]| = 1 - 2^{-q}$$

which is the maximum advantage that an adversary can get after $q$ queries. Hence, Fan et al.’s protocol has security flaws in ROM.

4 Security Analysis of SecLAP

Similar to their predecessors, Aghili et al.’s also claimed optimum security of SecLAP against an active adversary, where they argued security of their protocol
against different attacks informally and also formally evaluated its security using Burrows-Abadi-Needham (BAN) logic. In this section, we present the first third party evaluation of this protocol and show that, however, it is as insecure as its predecessor, i.e. Fan et al.’s protocol.

4.1 Partial secret disclosure attack

In this section, we present some properties of SecLAP that can be used to reveal information related to secret parameters. Given that information, the adversary can trace the tag or the reader which compromise their anonymity. The main observation is similar to the zero-sum property reported for Fan et al.’s protocol in Section 2. More precisely:

\[ \bigoplus_{j=0}^{2n-1} (M_{Rot(K_i)}(X,Y))_j = (\bigoplus_{j=0}^{n-1} od(M_{Rot(K_i)}(X,Y)))_j \bigoplus (\bigoplus_{j=0}^{n-1} ev(M_{Rot(K_i)}(X,Y)))_j \]

(4)

In the other hand, from the definition of \(M_{Rot(K_i)}(X,Y)\) we know that the \(od(M_{Rot(K_i)}(X,Y)) = (od(X)||ev(Y)) \oplus k_1 \ll [Y \oplus k_0 \mod n] \) and \(ev(M_{Rot(K_i)}(X,Y)) = (ev(X)||od(Y)) \oplus k_0 \ll [Y \oplus k_1 \mod n]. \) Hence:

\[ \bigoplus_{j=0}^{2n-1} (M_{Rot(K_i)}(X,Y))_j = (\bigoplus_{j=0}^{n-1}[(od(X)||ev(Y))\oplus k_1])_j \bigoplus (\bigoplus_{j=0}^{n-1}[(ev(X)||od(Y))\oplus k_0])_j \]

(5)

The above equation can be simplified as follow:

\[ \bigoplus_{j=0}^{2n-1} (M_{Rot(K_i)}(X,Y))_j = (\bigoplus_{j=0}^{n-1}X_j) \oplus (\bigoplus_{j=0}^{n-1}Y_j) \oplus (\bigoplus_{j=0}^{2n-1}(K_i)_j) \]

(6)

Next, assume that the adversary eavesdropped the first run of the exchanged messages between the reader and the tag, i.e., \(N_R \parallel M_1 \parallel M_2\), where \(M_1 = M_{Rot(K_i)}(TID \oplus N_R, od(K_i) \oplus N_T)\) and \(M_2 = M_{Rot(K_i)}(ev(K_i) \oplus N_R, TID \oplus N_T)\). Following the above argument, we can write:

\[ \bigoplus_{j=0}^{2n-1}(M_1)_j = (\bigoplus_{j=0}^{n-1}(TID \oplus N_R))_j \bigoplus (\bigoplus_{j=0}^{n-1}(od(K_i) \oplus N_T))_j \bigoplus (\bigoplus_{j=0}^{2n-1}(K_i)_j) \]

(7)

and:

\[ \bigoplus_{j=0}^{2n-1}(M_1)_j \bigoplus (\bigoplus_{j=0}^{n-1}(N_R)_j) \bigoplus (\bigoplus_{j=0}^{n-1}(N_T)_j) = (\bigoplus_{j=0}^{n-1}(TID))_j \bigoplus (\bigoplus_{j=0}^{n-1}(ev(K_i)))_j \]

(8)

Similarly, we can argue that:

\[ \bigoplus_{j=0}^{2n-1}(M_2)_j \bigoplus (\bigoplus_{j=0}^{n-1}(N_R)_j) \bigoplus (\bigoplus_{j=0}^{n-1}(N_T)_j) = (\bigoplus_{j=0}^{n-1}(TID))_j \bigoplus (\bigoplus_{j=0}^{n-1}(od(K_i)))_j \]

(9)

and, given Equations 8 and 9:

\[ (\bigoplus_{j=0}^{2n-1}(M_1)_j) \oplus (\bigoplus_{j=0}^{2n-1}(M_2)_j) = (\bigoplus_{j=0}^{2n-1}(K_i)_j) \]

(10)
Given that the secret parameters of SecLAP are $TID$, $RID$ and $K_i$, Equations 8 and 9 reveal two bits of the secrets. Given that as long as the tag has not updated its secrets those bits remain fixed, the adversary can use it as a source of traceability, which compromises the designers claim on the security of the protocol against traceability.

Given that the channel between the reader and the server is also insecure, to trace the reader, the adversary eavesdrops $IDX_i$, a sent message from the reader to the server, where $IDX_i = M\text{Rot}(K_i)(RID \oplus od(K_i), TID)$. Recall from Equation 6

$$\oplus_{j=0}^{2n-1} (IDX_i)_j = (\oplus_{j=0}^{n-1}(RID))_j \oplus (\oplus_{j=0}^{n-1}(TID))_j \oplus (\oplus_{j=0}^{n-1}(ev(K_i)))_j$$  \hspace{1cm} (11)

Combining Equations 11 and 8 reveals a single bit of $RID$ as follows:

$$(\oplus_{j=0}^{2n-1}(IDX_i)_j) \oplus (\oplus_{j=0}^{2n-1}(M_2))_j = \oplus_{j=0}^{n-1}(RID)_j$$  \hspace{1cm} (12)

It worth noting the related information in Equation 12 is independent of the tag’s data and also constant. Hence it is enough to compromise the reader anonymity.

Other messages that can be eavesdropped by the adversary are as follows, that can be used to disclose other information:

1. $IDC_i = M\text{Rot}(K_i)(ev(K_i) \oplus TID, RID)$, $M_3 = IDX_i \oplus K_i$ and $M_4 = M\text{Rot}(K_i)(N_T \oplus od(IDC_i), ev(IDC_i) \oplus N_R)$ sent from the reader to the server,
2. $M_5 = M\text{Rot}(K_i)(ev(K_i) \oplus N_S, od(IDC_i) \oplus N_R)$, $M_6 = M\text{Rot}(K_i)(od(K_i) \oplus N_S, ev(IDC_i) \oplus N_S)$ sent from the server to the reader,
3. $M_7 = M\text{Rot}(K_i)(od(K_i) \oplus N_S, TID \oplus N_T)$ and $M_8 = M\text{Rot}(K_i)(ev(K_i) \oplus N_S, TID \oplus N_S)$ sent from the reader to the tag,
4. $M_9 = M\text{Rot}(K_i)(od(K_i+1) \oplus TID, N_R \oplus N_S)$ and $M_{10} = M\text{Rot}(K_i)(od(K_i+1) \oplus TID \oplus ev(K_i+1), N_S)$ sent from the tag to the reader,
5. $M_{11} = M\text{Rot}(K_i)(ev(IDC_i) \oplus ev(K_i+1), N_R \oplus N_S)$, $M_{12} = M\text{Rot}(K_i)(od(IDC_i) \oplus ed(K_i+1), K_r \oplus N_S)$, $IDX_{i+1} = M\text{Rot}(K_i)(RID \oplus od(K_i+1), TID)$, $IDC_{i+1} = M\text{Rot}(K_i)(TID \oplus ev(K_i+1), RID)$, $M_{13} = IDX_{i+1} \oplus K_i$, $M_{14} = M\text{Rot}(K_i)(ev(IDC_{i+1}) \oplus M_{11}, N_S \oplus N_S)$, $M_{16} = M\text{Rot}(K_i)(od(IDC_{i+1}) \oplus ev(IDC_{i+1}), N_S)$ sent from the reader to the server,
6. $M_{17} = M\text{Rot}(K_i)(od(K_i+1) \oplus ev(IDC_{i+1}), N_S \oplus N_R)$ sent from the server to the reader,
7. $M_{18} = M\text{Rot}(K_i)(ev(K_i+1) \oplus N_T \oplus od(K_i+1), TID \oplus N_S)$ sent from the reader to the tag.

For example:

$$(\oplus_{j=0}^{2n-1}(M_6)_j) \oplus (\oplus_{j=0}^{2n-1}(IDC_i)_j) = (\oplus_{j=0}^{n-1}(ev(K_i)))_j$$  \hspace{1cm} (13)

Now combining Equations 8 and 13 reveals a bit of $TID$, as follows:

$$(\oplus_{j=0}^{2n-1}(M_6)_j) \oplus (\oplus_{j=0}^{2n-1}(IDC_i)_j) \oplus (\oplus_{j=0}^{n-1}(M_1)_j) = (\oplus_{j=0}^{n-1}(ev(K_i)))_j \oplus (\oplus_{j=0}^{n-1}(ev(K_i)))_j \oplus (\oplus_{j=0}^{n-1}(ev(K_i)))_j = (\oplus_{j=0}^{n-1}(TID)_j$$
Given that \( TID \) will not be updated at the end of the session, it can be used to compromise the tag’s anonymity and trace the tag holder in any session.

Similarly,

\[
(\oplus_{j=0}^{2n-1}(M_9)_j) \oplus (\oplus_{j=0}^{2n-1}(M_{10})_j) \oplus (\oplus_{j=0}^{2n-1}(N_R)_j) = (\oplus_{j=0}^{n-1}(ev(K_{i+1}))_j)
\]  

(15)

and from \( IDX_{i+1} \) and Equation 10, 12 and 14 we can disclose \((\oplus_{j=0}^{n-1}(od(K_{i+1}))_j)\), which combined with Equation 15 reveals two bits of \( K_{i+1} \) and also \((\oplus_{j=0}^{n-1}(K_{i+1}))_j\).

The success probability of disclosing all values mentioned in this section is ‘1’ and the complexity is only eavesdropping one session of the protocol.

4.2 Full secret disclosure attack

The secret parameters of SecLAP are \( K_i \in \{0,1\}^{2n} \), \( TID \in \{0,1\}^n \) and \( RID \in \{0,1\}^n \), where \( TID \) and \( RID \) are constant values while \( K_i \) is a dynamic value which is updated after each successful run of the protocol. Most of the transferred messages are masked by \( K_i \) and produced by \( MRot(K_i)(.) \), which is more complicated than \( Cro(.) \) which has been used by Fan et al. Hence, designers of SecLAP expect a good security of the protocol against secret disclosure attack. However, we present an attack to extract those secret parameters efficiently. Despite of the designers claim that SecLAP is secure even against an active adversary who has full control over the channel between the tag and the reader and the reader and the server, we consider the weakest adversary who can just eavesdrop the channel between the tag and the reader and even not the server and the reader.

To start the attack, we assume that the adversary eavesdrops any transferred message form the tag \( T \) to the reader \( R \) or from the \( R \) to \( T \) in session \( i \), which we call this phase of attack the learning phase of attack. Hence, at the end of the learning phase of the attack, the adversary has the below information:

- \( N_R \) sent from \( R \) to \( T \);
- \( M_1 = MRot(K_i)(TID \oplus N_R, od(K_i) \oplus N_T) \) and \( M_2 = MRot(K_i)(ev(K_i) \oplus N_R, TID \oplus N_T) \) sent from \( T \) to \( R \);
- \( M_7 = MRot(K_i)(od(K_i) \oplus N_S, TID \oplus N_T) \) and \( M_8 = MRot(K_i)(ev(K_i) \oplus N_S, TID \oplus N_T) \) sent from \( R \) to \( T \);
- \( M_9 = MRot(K_i)(od(K_{i+1}) \oplus TID, N_R \oplus N_S) \) and \( M_{10} = MRot(K_i)(od(K_{i+1}) \oplus TID \oplus ev(K_{i+1}), N_S) \) sent from \( T \) to \( R \);
- \( M_{18} = MRot(K_i)(ev(K_{i+1}) \oplus N_T \oplus od(K_{i+1}), TID \oplus N_S) \) sent from \( R \) to \( T \).

On the other hand, from the definition of \( MRot(K_i)(X,Y) \) we know that the \( od(MRot(K_i)(X,Y)) = [(od(X)||ev(Y)) \oplus k_1] \equiv [(Y \oplus k_0) \mod n] \) while \( ev(MRot(K_i)(X,Y)) = [(ev(X)||od(Y)) \oplus k_0] \equiv [(Y \oplus k_1) \mod n] \) is used to produce the even bits of \( MRot(K_i)(X,Y) \). Hence, we can argue that any bit \( r \) of the output, \( O_r = (MRot(K_i)(X,Y)_r) \), is a linear function of a bit from \( K \) with either a bit from \( X \) or a bit from \( Y \), i.e. \( O_r = c \times (X)_r \oplus \bar{c} \times (Y)_r \oplus (K_b)_u \), where \( b,c \in \{0,1\} \), \( r \in \{0,1,\ldots,2n-1\} \), and \( s,t,u \in \{0,1,\ldots,n-1\} \). Although we may not know the exact value of \( s,t \) and \( u \). However, again from
the definition of $M\text{Rot}((K_i))(X,Y)$ we know that if $r$ is an odd value then $s$ will also be an odd value while $t$ will be even and if $r$ is an even value then $s$ will also be even while $t$ will be odd. In addition, if $O_r = (X)_z \oplus (Y)_z \oplus (K_b)_z$ then $O_{r+2z} = (X)_{z+2} \oplus (Y)_{z+2} \oplus (K_b)_{z+2}$, for any $z \in \{0, 1, \ldots, n - 1\}$, obviously addition takes place modulo $n$. In addition, given $u$ and $b$ in $(k_b)_u$, we can uniquely determine related values of $s$ and $t$. It comes from the fact that $od(M\text{Rot}((K_i))(X,Y)) \Rightarrow [(Y \oplus k_0) \mod n] \equiv [(od(X)||ev(Y)) \oplus k_1]$ while $ev(M\text{Rot}((K_i))(X,Y)) \Rightarrow [(Y \oplus k_1) \mod n] \equiv [(ev(X)||od(Y)) \oplus k_0]$. Therefore, given to extract 2n-linear equation out of $M\text{Rot}((K_i))(X,Y)$, the only unknown parameter will be $[(Y \oplus k_1) \mod n]$ and $[(Y \oplus k_0) \mod n]$, which has the total complexity of $n + n = 2n$. In the rest of the paper we use odd-offset of and even-offset $ef$ to denote $[(Y \oplus k_1) \mod n]$ and $[(Y \oplus k_0) \mod n]$ respectively. Now, given the eavesdropped $M_1$, $M_2$, $N_R$ and $N_T$ from a session of the protocol and the above properties of $M\text{Rot}((K_i))(X,Y)$, the adversary uses the below procedure to extract the secret parameters of SecLAP:

1. for $of1 = 0, \ldots, n - 1$:
   (a) $od(M_1) \Rightarrow of1 \rightarrow [(od(TID) \oplus N_R)||ev(od(K_i) \oplus N_T)] \oplus k_0$;
   (b) $ev(M_1) \Rightarrow ef1 \rightarrow [(ev(TID) \oplus N_R)||od(K_i) \oplus N_T)] \oplus k_1$;
   (c) for $of2 = 0, \ldots, n - 1$:
     (d) for $ef2 = 0, \ldots, n - 1$:
       i. $od(M_2) \Rightarrow of2 \rightarrow [(od(K_i) \oplus N_R)||ev(TID \oplus N_T)] \oplus k_0$;
       ii. $ev(M_2) \Rightarrow ef2 \rightarrow [(ev(K_i) \oplus N_R)||od(TID \oplus N_T)] \oplus k_1$;
       iii. Steps $[2a]$, $[2b]$, $[2d]i$ and $[2d]ii$ produce 4n linear equations while the unknown parameters are $TID \in \{0, 1\}^n$ and $K_i \in \{0, 1\}^{2n}$. Hence, the adversary uses 3n linearly independent equations to determine $TID$ and $K_i$ and the rest of equations to filter wrong guesses.

iv. return the candidate $TID$ and $K_i$ of $of1$, $ef1$, $of2$ and $ef2$.

3. Given that $of1 = [(od(K_i) \oplus N_T) \oplus k_1] \mod n$, $ef1 = [(od(K_i) \oplus N_T) \oplus k_0] \mod n$, $of2 = [(TID \oplus N_T) \oplus k_1] \mod n$ and $ef2 = [(TID \oplus N_T) \oplus k_0] \mod n$ the returned $TID$ and $K_i$ should also pass.

The above attack has the complexity of $4n$ time solving a linear equation with $4n$ equations of $3n$ independent variable. Hence, it is expected to do not return any candidate exclude the correct pair of $TID$ and $K_i$ in Step $[2d]iv$. However, any possible wrong guess also filtered in Step $[3]$ Given that any wrong guess passes Step $[3]$ with the probability of $n^{-4}$, the algorithm will return the correct value of $TID$ and $K_i$. Next, given $TID$ and $K_i$ and also the eavesdropped $M_T$ the adversary extracts $N_S$ and calculates $K_{i+1} = M\text{Rot}((K_i))(od(K_i) \oplus N_S \oplus N_R \oplus N_T \oplus ev(K_i),TID)$. The adversary can also use other eavesdropped messages, i.e., $M_8$, $M_9$, $M_{10}$ and $M_{18}$ to filter any possible wrong guess.

Given that the computational complexity of solving a system of linear equation of $N$ variables has the complexity of $O(N^3)$, the expected complexity of extracting $TID$, $K_i$ and $K_{i+1}$ is $n^4(3n)^3 = 27n^7$. For $n = 128 = 2^7$, the adversary will be able to extract those secret parameters with the complexity of $O(2^{54})$, while the expected complexity is at least $2^{128}$, which shows a huge gape.
It should be noted, given that the channel between the reader and the server is insecure also, the adversary can eavesdrop $IDX_i = MRot(K_i)(RID \oplus od(K_i), TID)$. Assuming that the adversary has already disclosed $TID$ and $K_i$, it can easily extract $RID$ also. Similarly, the adversary can use $M_7$ to extract $N_s$ which, along with other known parameters, can be used to construct $K_{i+1} = MRot(K_i)(od(K_i) \oplus N_S \oplus N_R \oplus N_T \oplus ev(K_i), TID)$. In this way, the adversary revealed whole secret parameters of the protocol with the computation complexity of $27n^2$ and just eavesdropping one session of the protocol between the protocol parties, i.e. tag, reader and server.

4.3 Traceability attack

In Section 4.2, we presented an attack where any passive adversary who eavesdrops the transferred messages of a session of the protocol between the legitimate tag and the reader and server will be able to extract $TID$, $RID$, $K_i$ and $K_{i+1}$ with the complexity of $O(2^{54})$. Given that $TID$ and $RID$ are constant values for any tag/reader and will not be updated after the protocol completion, they can be used to trace the tag/reader, which contradicts the designers claim.

4.4 Distinguishability in ROM

In this section, similar to the case of Fan et al., we show that SecLAP is not also a secure protocol in ROM. To distinguish SecLAP from ROM we use the observation used to partial recover secret parameters in Section 4.1. From the structure of the messages one can deduce that:

\[ (\oplus_{j=0}^{2n-1} (M_1)_j) \oplus (\oplus_{j=0}^{2n-1} (M_2)_j) \oplus (\oplus_{j=0}^{2n-1} (M_3)_j) \oplus (\oplus_{j=0}^{2n-1} (IDX_i)_j) = 0 \quad (16) \]

This property can be used as a metric to determine that the used protocol is SecLAP and not \( R \).

To distinguish SecLAP \( P \) from \( R_m \), the adversary does as follows:

1. eavesdrops messages of a session of the given protocol, transferred over channel.
2. evaluates Equation 16 and returns \( P \) if it is true; otherwise returns \( R_m \).

To determine the adversary’s advantage, it is clear if the adversary communicates with SecLAP then with the probability of ‘1’ returns \( P \) in Step 2, while if it communicates with \( R_m \) then returns \( P \) with the probability of ‘2^{-1}'. To improve the adversary’s advantage, we can increase the number of eavesdropped sessions. For an adversary who eavesdropped \( q \) sessions, the advantage will be as follows:

\[ |Pr[D^C = 1] - Pr[D^{R_m} = 1]| = 1 - 2^{-q} \] which is the maximum advantage that an adversary can get after \( q \) queries. It should be noted other combination of transferred messages can also be used for distinguishing SecLAP from ROM. For example:
\[ (\oplus_{j=0}^{2n-1} (M_1)_j) \oplus (\oplus_{j=0}^{2n-1} (M_2)_j) \oplus (\oplus_{j=0}^{2n-1} (M_T)_j) \oplus (\oplus_{j=0}^{2n-1} (M_8)_j) \oplus (\oplus_{j=0}^{2n-1} (N_T)_j) = 0 \] (17)

An interesting point with Equation (17) is the fact to distinguish the protocol the adversary only requires to eavesdrop the channel between the tag and the reader. It is clear that the adversary can use both Equation 17 and 16 (and several other combinations) to achieve same advantage with eavesdropping less number of sessions.

Therefore, similar to Fan et al.’s protocol, SecLAP has security flaws in ROM.

5 Discussion and Conclusion

In the previous sections, we showed that the Fan et al.’s protocol and its successor SecLAP do not provide the desired security against various attacks. More interestingly, all presented attacks work in a passive adversary model and have very low data/time complexity and the success probability of 1, exclude the full secret disclosure attack against SecLAP which has the complexity of $27n^7$ which also is very low compared to the secret key length which is $2n$-bit. Although these protocols have trivial flaws and they could be designed more sophisticated, such that will not be compromised that badly, however, previous studies show that it may not be possible to design a secure authentication protocol just using several calls to ultra-lightweight operations such as the Cro(.) function which has been designed by Fan et al. and MRot$(K_i)(.)$ function which has been designed by Aghili et al. It should be noted, designing a protocol from this point of view has a long unsuccessful history. SASI [11], RAPP [22], SLAP [15] and LMAP [17] are just examples that have been compromised by the following analysis [4,3,19,21]. Hence, attempting to improve the current protocols by keeping their basic structure and just improving the way that messages are calculated may not be possible and would be just proposing another easy to break protocol. Hence, we avoid designing the improved version of this protocol in this paper. However, our suggestion is to consider the recent advances in lightweight cryptography in designing new protocols. For example, many lightweight block ciphers have been proposed in recent years that can be implemented even in passive RFID tags, based on their reported implementation results. SKINNY [7], SIMON [6], SIMECK [26] and Midori [5] are just examples. In addition, the current ongoing NIST competition for lightweight cryptography [12] also announced its first-round candidates, which aims to provide secure building blocks for constrained environments such as RFID and IoT. Hence as future work, we suggest to design a secure protocol for mutual authentication based on these schemes rather than developing a new ultra-lightweight function, such as Fan et al.’s Cro(.) function or SecLAP’s MRot$(K_i)(.)$ function, and designing an easily breakable protocol based on it.

It should be noted, the Fan et al.’s protocol and also its successor SecLAP are not efficient protocols in the term of the number of the messages transferred between the tag and the reader, which are 7 and 5 runs respectively. It should be
possible to design a protocol with much less number of messages that achieves all security targets of their protocol. This can be also considered as a target for future works.

References