SRMAP and ISLAP Authentication Protocols: Attacks and Improvements

Mohammad Mardani Shahrbabak[a], Shahab Abdolmaleky[b,*]

[a] *Assistant Professor, Imam Hossein Comprehensive University, Tehran, Iran.*
[b] *Department of Computer Engineering, Science and Research Branch, IAU University Department, Tehran, Iran.*

Keywords:

RFID authentication protocol, Security attacks, Privacy attacks, Traceability, EPC C1 G2 standard

ABSTRACT

RFID technology is a system which uses radio frequency to transmit data. Data transmission between Tags and Readers is wireless which can be easily eavesdropped by adversary. Due to security and privacy reasons, various authentication protocols proposed. In this paper, we cryptanalyze two different RFID authentication protocols and it is shown that either of them have some weaknesses. In 2014, *Chang et al.* proposed a mutual authentication protocol for RFID technology based on EPC Class 1 Generation 2 standard. We show that their protocol is not safe regard to privacy and cannot repulse neither *Traceability attack* nor *Forward Traceability attack*. Also, in 2015, *Pourpouneh et al.* proposed a server-less authentication protocol. We discover that their protocol is not able to thwart security and privacy attacks such as *Secret Parameter Reveal, Traceability and Forward Traceability*. In addition, we robust the two schemes to defend those attacks which can protect RFID users against different threats. Then, analyzing of the protocols are compared with some state-of-art ones.

1 Introduction

RFID which stands for Radio Frequency identification is a technology, which can capture data and process transaction in wireless. Using RFID technology have been changed dramatically over the past few years. These days, RFID technology is using in many domains such as Government Library, People Tracking, Manufacturing & Aerospace, Healthcare and etc. [1]. Most of the time in libraries, stuffs is registering data of books which are borrowed or currently bought. Using RFID in libraries can automatize the tasks is doing by stuff members and safes their time [2]. RFID is also used to track persons, in public transportation systems we can automatically track every passenger in real time [3]. RFID systems consists of three major sections. A backend server, readers and tags. The reader gets the data form an RFID tag which describe carrier of tag. Then the reader transmit data to the backend server and check whether the tag is legitimate. Tags are authenticated in this way. By cause of RFID usage in sensitive domains, it is very serious to provide secure authentication. In general, RFID authentication protocols are analyzing base on formal and informal methods. Many privacy model have been proposed which are based on formal methods [4-11] and informal methods which also known as *ad-hoc* methods are [12-16]. In order to provide security and privacy, various authentication protocols have been proposed for RFID systems [17-19].

In 2014, Chang et al.'s cryptanalyzed Cho et al.'s protocol [20] and proposed an improved version of it, which is claim to be secure in terms of security. In [14], Aref et al. show that [21] is not secure and it suffers to omit some practical security attacks. In this study, we show that Chang et al.'s protocol (referred as SRMAP) cannot resists privacy attacks such as Traceability and Forward Traceability attacks. Even though, we modify the protocol to eliminate all of the privacy weaknesses.

In 2014, Deng et al. proposed a server-less authentication protocol which is an improved version of Hoque et al. protocol [22]. Subsequently in 2015, Abdolmaleky et al. elucidate that Deng et al.'s protocol is not secure against *Secret parameter reveal*, *Tag impersonation* and *Reader impersonation* attacks [23]. Also, in 2015, Pourpouneh et al. show that Deng et al.'s protocol is not safe and suffer from data desynchronization attack [24]. In addition, they proposed an improved version of Deng et al.'s protocol which is

*Corresponding author.
E-mail addresses: sh.abdolmaleky@srbiau.ac.ir (S. Abdolmaleky)
, mmardani@ihu.ac.ir (M. Mardanishahrbabak).

claim to be secure. Also, they claim that the protocol can resists against privacy attacks. In this study, the Pourpouneh et al.'s protocol (referred as ISLAP) is cryptanalyzed and it is shown that their protocol suffer from security and privacy deficiency and has another drawbacks which cannot resists Secret parameter reveal, Traceability and Forward traceability attacks. The cost of Secret parameter reveal attack is maximum 2^L computations. In order to overcome the attacks as highlighted previously, we propose an enhanced version of the ISLAP protocol which is able to thwart all the mentioned attacks.

The structure of paper is organized as follows: In section 2, Ouafi and Phan model is described. SRMAP protocol is analyzed and also we show weaknesses in their protocol in section 3. In section 4, we describe and investigate vulnerabilities of the ISLAP protocol. Then, in section 5, we propose an improved version over both SRMAP and ISLAP. Finally, in section 5, the paper is concluded.

2 Formal privacy model of Ouafi and Phan

Ouafi and *Phan* introduced a privacy model to evaluate privacy of RFID protocols [8]. We use *Ouafi-Phan* privacy model to analyze the protocols, so the model is summarized as follows.

The attacker $\mathcal{A}$ is able to listen on whole channels between tags and readers and also she can attack them in active and passive mode. Besides, the following queries is grant to the attacker $\mathcal{A}$ to execute:

- ***Execute query*** $(\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{i})$***:*** The attacker can eavesdrop and obtain all transmitted messages between the tag,T, and the reader,R, in ith session. This query cause passive attacks.
- ***Send query*** $(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{m}, \boldsymbol{i})$***:*** The attacker $\mathcal{A}$ has permission to impersonate a reader U within the ith session, and forwards a message m to a tag V. Also, the attacker $\mathcal{A}$ has permission to alert or block the exchanged message m between the tag and the reader. Note that U and V are members of readers and tags sets, respectively. This query cause active attacks.
- ***Corrupt query*** $(\boldsymbol{T}, \boldsymbol{K'})$***:*** The attacker $\mathcal{A}$ has permission to access secret parameters of the tag, T. Indeed, the attacker $\mathcal{A}$ has physical access to the tag database and can set the secret parameter K'.
- ***Test query*** $(\boldsymbol{T_0}, \boldsymbol{T_1}, \boldsymbol{i})$***:*** When this query is executed in the particular session i, after finishing the ith session, a random number bit, $b \epsilon \{0,1\}$ is generated by challenger and delivered $T_b \epsilon \{T_0, T_1\}$ to the attacker. Now, the attacker succeeds if she can guess the bit b correctly.

Untraceable privacy (UPriv): Untraceable privacy is defined by the game G that is played between an attacker $\mathcal{A}$ and a set of the tags and the reader. It means, an attacker $\mathcal{A}$ plays game G using collected information of the reader and the tag. The game G can be played using mentioned queries as follows.

- ***Learning phase:*** The attacker $\mathcal{A}$ is able to send any of the queries such as *Execute*, *Send* and *Corrupt*. Also, she can transmit data with the reader R and T_0, T_1 which selected randomly.
- ***Challenge phase:*** The attacker $\mathcal{A}$ selects two tags,$\mathrm{T}_0$and T_1, and forwards a $Test\ query(T_0, T_1, i)$ to the challenger. Then, the challenger selects a bit $b \epsilon \{0,1\}$ randomly and the attacker $\mathcal{A}$ determines the tag $\mathrm{T}_b \in \{\mathrm{T}_0, \mathrm{T}_1\}$ using *Execute* and *Send* queries.
- ***Guess phase:*** Eventually, the attacker $\mathcal{A}$ finishes the game G and returns a bit $b' \epsilon \{0,1\}$as a guess of b.

The success of attacker $\mathcal{A}$ in game G and consequently breaking the notion of *UPriv* is measured by $\mathcal{A}$'s advantage in recognizing whether the attacker $\mathcal{A}$ received T_0 or T_1, and denoted by $\mathrm{Adv}_{\mathcal{A}}^{UPriv}(k)$ where k is the security parameter.

$$\mathrm{Adv}_{\mathcal{A}}^{UPriv}(k) = |\mathrm{pr}(b' = b) - \mathrm{pr}(\text{random coin flip})| = \left|\mathrm{pr}(b' = b) - \frac{1}{2}\right|$$

where $0 \leq \mathrm{Adv}_{\mathcal{A}}^{UPriv}(k) \leq \frac{1}{2}$. Note that, if $\mathrm{Adv}_{\mathcal{A}}^{UPriv}(k) \ll \varepsilon(k)$, the protocol is traceable with negligible probability.

In the rest of paper, we use *Ouafi-Phan* privacy model for our privacy analysis of ISLAP and SRMAP protocols.

3 Analysis of SRMAP Protocol

This section aims to show the privacy of *SRMAP* protocol have some problems and suffer it from attacks such as traceability attack and forward traceability attacks. To this aim, *SRMAP* protocol is described firstly [21].

A. SRMAP Protocol

The notations, which are used in the protocol, illustrated in Table I.

TABLE I. THE NOTATIONS OF SRMAP PROTOCOL

Notation	Description
$\mathbf{EPC}_s$	Electronic Product Code
$\mathbf{I_j}$	The database index stored in the tag to find the corresponding record of the tag in the reader.
$\mathbf{Rid_k}$	The identification code of kth reader.
$\mathbf{Auk_j}$	The authentication key stored in the tag to be used by the back-end server to authenticate the tag at the $(i+1)$th authentication phase.
$\mathbf{Ack_j}$	The access key stored in the tag to be used by back-end server to authenticate the tag at the $(i+1)$th authentication phase.
DATA	The corresponding data for the tag kept in the back-end server.
$\mathbf{H}(.)$	Hash function.
$\mathbf{PRNG}(.)$	Pseudo random number generator
$\mathbf{A \oplus B}$	Message A is XORed with message B

The structure of *SRMAP* protocol that is shown in Fig. 1 is summarized as follows,

a) Initial phase

In this phase, authentication key Auk_1, access key Ack_1 and index value I_1 for each tag are generated by server. The server keeps a set of data for each tag, $\{Ack_{old}, Auk_{old},\ I_{old},\ Ack_{new},\ Auk_{new},\ I_{new},\ Rid_k,\ EPC,\ data\}$ where Auk_{old} used in previous session and Auk_{new} is used in currently shared with the tag.

b) Authentication phase

This phase includes six steps as follows,

Step 1. Reader → Tag: The reader generates R_r as a random number and sends it to the tag.

Step 2. Tag → Reader: Once receiving R_r, the tag generates a random number R_t. It computes the following messages then sends R_t, M_1 and I_j to the reader.

$$M_1 = Auk_j \oplus I_j \oplus PRNG(EPC_s \oplus Ack_j \oplus R_r \oplus R_t)$$

Step 3. Reader → Server: The reader calculates $A = H(Rid_k \oplus R_r)$ and forwards the messages (M_1, R_t, I_j, A, R_r) to the server.

Step 4. Server → Reader: Based on the received messages from the reader, the server performs the following operations,

a) The server verifies $H(Rid_k \oplus R_r) \overset{?}{=} A$ with R_r and follows the rest of authentication procedure.

b) The server uses the received I_j to search the tag. If the tag does not exists, the server terminates authentication operation, otherwise it goes to next steps.

c) Now by using the obtained data of the tag, the server verify $Auk_{new} \oplus I_{new} \oplus PRNG(EPC_s \oplus Ack_{new} \oplus R_r \oplus R_t \overset{?}{=} M_1$. If the verification succeed, the tag is authenticated; otherwise the server aborts the protocol. Then the server uses the identity of the reader to hide the $data_i$ of the tag as $B = data_i \oplus Rid_k$. Then the server computes the following messages and updates all keys of the tag and responds to the reader by messages B, C and M_2.

$$M_2 = PRNG(Auk_{new} \oplus R_t) \oplus Ack_{new}$$

$$C = H(data_i \oplus R_r)$$

$$Ack_{old} \leftarrow Ack_{new} \leftarrow PRNG(Ack_{new})$$
$$Auk_{old} \leftarrow Auk_{new} \leftarrow PRNG(Auk_{new})$$
$$I_{old} \leftarrow I_{new} \leftarrow PRNG(Ack_{new} \oplus I_{new})$$

Step 5. Reader → Tag: Now using the received message from the server, the reader verifies $H(data_i \oplus R_r) \overset{?}{=} C$. If the verification passed then sends M_2 to the tag.

Step 6. Utilizing the received message M_2, the tag uses its keys to verifies $M_2 \overset{?}{=} PRNG\big(Auk_j \oplus R_t\big) \oplus Ack_j$. If the verification passed, the tag computes $Ack_{j+1} \leftarrow PRNG\big(Ack_j\big)$, $Auk_{j+1} \leftarrow PRNG\big(Auk_j\big)$ and $I_{j+1} \leftarrow PRNG\big(Ack_j \oplus I_j\big)$. Eventually, the tag replaces Auk_j, Ack_j and I_j with Auk_{j+1}, Ack_{j+1} and I_{j+1} respectively, for another session.

B. Traceability Attack

One of the main weaknesses of *SRMAP* protocol is the fact that the tag updates its parameter I_j, after a successful authentication. Thus the attacker can use this weakness and trace a tag. This attack can be summarized as follows,

Learning phase: In round (i), the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$ to the tag, and obtains $I_j^{T_0}$ after that the attacker $\mathcal{A}$ sends an $Send\ query(R, T_0, i)$, and

Database $(Ack_{old}, Auk_{old}, I_{old}, Ack_{new}, Auk_{new}, I_{new}, Rid_k, EPC, data)$	**Reader** (Rid_k)	**Tag** $\big(Ack_j, Auk_j, I_j, EPC_s\big)$
$For\ each\ Rid_k in\ Database\ verify\ H(Rid_k \oplus R_r) \overset{?}{=} A$ $If\ \ I_{new} = I_j :\ X = new$ Else $I_{old} = I_j : X = old$ End $Verify\ \ Auk_{new} \oplus I_{new} \oplus PRNG(EPC_s \oplus Ack_{new} \oplus R_r \oplus R_t \overset{?}{=} M_1$	$R_r \rightarrow$ $A = H(Rid_k \oplus R_r)$ $\leftarrow \big(M_1, R_t, I_j\ , R_r, A\big)$	Generates R_t $M_1 = Auk_j \oplus I_j \oplus PRNG(EPC_s \oplus Ack_j \oplus R_r \oplus R_t)$ $\leftarrow \big(M_1, R_t\ I_j\big)$
Then computes: $M_2 =\ PRNG\big(Auk_j \oplus R_t\big) \oplus Ack_j$ $B = data_i \oplus Rid_k$ $C = H(data_i \oplus R_r)$ $If\ \ X = new$ $Ack_{old} \leftarrow Ack_{new} \leftarrow PRNG\big(Ack_j\big)$ $Auk_{old} \leftarrow Auk_{new} \leftarrow PRNG\big(Auk_j\big)$ $I_{old} \leftarrow I_{new} \leftarrow PRNG\big(Ack_j \oplus I_j\big)$ $Else$ Do nothing End If $(M_2, B, C) \rightarrow$	$data_i = B \oplus Rid_k$ $Verify\ \ H(data_i \oplus R_r) \overset{?}{=} C$ $M_2 \rightarrow$	$Verify\ M_2 \overset{?}{=} PRNG\big(Auk_j \oplus R_t\big) \oplus Ack_j$ $Ack_{j+1} \leftarrow PRNG\big(Ack_j\big)$ $Auk_{j+1} \leftarrow PRNG\big(Auk_j\big)$ $I_{j+1} \leftarrow PRNG\big(Ack_j \oplus I_j\big)$

Fig. 1. SRMAP protocol [21].

blocks protocol. As a result the tag does not updating secret values.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(T_0, T_1, i+1)$. According to the randomly chosen bit $b \epsilon \{0,1\}$, the attacker is given a tag $T_b \epsilon \{T_0, T_1\}$. After that, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+1)$ by sending N_1 message, and obtains $I_{j+1}^{T_b}$.

Guess phase: Eventually, the attacker $\mathcal{A}$ stops the game G, and outputs a bit $b' \epsilon \{0,1\}$ as a guess of bit b as follows.

$$b' = \begin{cases} 0 & if \quad I_{j+1}^{T_b} = I_j^{T_0} \\ 1 & otherwise \end{cases}$$

As a result, it can be written:

$$Adv_A^{upriv}(K) = |pr(b' = b) - pr(random\ coin\ flip)|$$
$$= \left|pr(b' = b) - \tfrac{1}{2}\right| = \left|1 - \tfrac{1}{2}\right| = \tfrac{1}{2} \gg \varepsilon$$

Proof: After an unsuccessful challenge between the attacker and the tag, the tag does not update $I_j^{T_0}$. As a result, the tag uses the same value in the next session.

C. Forward Traceability Attack

In this section, it is shown that *SRMAP et al.*'s protocol also does not assure the forward traceability. According to the configuration of *SRMAP et al.*'s protocol, we observe that the secret values ID_t and k are fixed in all rounds. Using this issue, an attacker can trace the target tag as follows.

Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $\left(EPC_{s,i}^{T_0}, Auk_j^{T_0}, Ack_j^{T_0}, I_j^{T_0}\right)$ from tag T_0. Now the attacker can compute $Auk_{j+2}^{T_0}$ and $Ack_{j+2}^{T_0}$ at the session $j+2$ by two times repeating $PRNG$ of $Auk_j^{T_0}$and $Ack_j^{T_0}$. Consequently, $I_{j+2}^{T_0}$ can computes as $I_{j+2}^{T_0} = PRNG\left(PRNG\left(Ack_j^{T_0}\right) \oplus PRNG\left(I_j^{T_0} \oplus Ack_j^{T_0}\right)\right)$.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(\ T_0, T_1, j)$. According to the randomly chosen bit $b \epsilon \{0,1\}$, the attacker is given a tag $T_b \epsilon \{T_0, T_1\}$. After that, in round $(j+2)$th, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, j+2)$ by sending $R_{r,j}$ (i.e., the same value as for session i) and obtains $\left(M_{1,j+2}^{T_b}, R_{t,j+2}^{T_b}\right)$.

Guess phase: The attacker $\mathcal{A}$ stops the game *G*, and outputs a bit $b' \epsilon \{0,1\}$ as a guess of bit b. In order to guess b', firstly the attacker $\mathcal{A}$ computes $\alpha = I_{j+2}^{T_0} = PRNG\left(PRNG\left(Ack_j^{T_0}\right) \oplus PRNG\left(I_j^{T_0} \oplus Ack_j^{T_0}\right)\right)$, $\beta = Ack_{j+2}^{T_0} = PRNG\left(PRNG\left(Ack_j^{T_0}\right)\right)$, $\gamma = Auk_{j+2}^{T_0} = PRNG\left(PRNG\left(Auk_j^{T_0}\right)\right)$ and $\chi = PRNG\left(EPC_{s,j}^{T_0} \oplus \beta \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right)$.Then, outputs a bit $b^{'} \epsilon \{0,1\}$ as a guess of bit b using the following rule.

$$b' = \begin{cases} 0 & if \ \ M_{1,j+2}^{T_b} \oplus \gamma \oplus \alpha = \chi \\ 1 & otherwise \end{cases}$$

As a result, it can be written that,

$$Adv_A^{upriv}(K) = |pr(b' = b) - pr(random\ coin\ flip)|$$
$$= \left|pr(b' = b) - \tfrac{1}{2}\right| = \left|1 - \tfrac{1}{2}\right| = \tfrac{1}{2} \gg \varepsilon$$

It is clear that base on the mentioned attack, the attacker can obtain $Auk_{j+n}^{T_0}$and $Ack_{j+n}^{T_0}$ for $n \geq 1$ using obtain $Auk_j^{T_0}$ and $Ack_j^{T_0}$.

Proof: Since the value of EPC_s is fixed in all rounds, thus $EPC_{s,j}^{T_0} = EPC_{s,j+2}^{T_0}$. Using this fact, the following equations can be written.

$If\ \ T_b = T_0$

$$M_{1,j+2}^{T_b} \oplus \gamma \oplus \alpha = Auk_{j+2}^{T_b} \oplus I_{j+2}^{T_b} \oplus PRNG\left(EPC_{s,j+2}^{T_b} \oplus Ack_{j+2}^{T_b} \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right) \oplus PRNG\left(PRNG\left(Auk_j^{T_0}\right)\right) \oplus PRNG\left(PRNG\left(Ack_j^{T_0}\right) \oplus PRNG\left(I_j^{T_0} \oplus Ack_j^{T_0}\right)\right)$$

It can be seen that $I_{j+2}^{T_0} = PRNG\left(PRNG\left(Ack_j^{T_0}\right) \oplus PRNG\left(I_j^{T_0} \oplus Ack_j^{T_0}\right)\right)$ and $Auk_{j+2}^{T_0} = PRNG\left(PRNG\left(Auk_j^{T_0}\right)\right)$, Eq. (1) can be rewritten as follows,

$$= Auk_{j+2}^{T_b} \oplus I_{j+2}^{T_b} \oplus PRNG\left(EPC_{s,j+2}^{T_b} \oplus Ack_{j+2}^{T_b} \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right) \oplus Auk_{j+2}^{T_0} \oplus I_{j+2}^{T_0}$$

Then, with the suppose of $T_b = T_0$, Eq. (2) can be rewritten as follows,

$$= PRNG\left(EPC_{s,j+2}^{T_0} \oplus Ack_{j+2}^{T_0} \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right)$$

By using the value of $Ack_{j+2}^{T_0} = PRNG\left(PRNG\left(Ack_j^{T_0}\right)\right) = \beta$, Eq. (2) can be rewritten as follows,

$$= PRNG\left(EPC_{s,j+2}^{T_0} \oplus \beta \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right)$$

finally, As we know $EPC_{s,j+2}^{T_0} = EPC_{s,j}^{T_0}$, it can be written as:

$$= PRNG\left(EPC_{s,j+2}^{T_0} \oplus \beta \oplus R_{r,j} \oplus R_{t,j+2}^{T_b}\right)$$
$$= \chi.$$

4 Analysis of ISLAP Protocol

This section aims to show that the security and the privacy of *ISLAP* protocol have some weaknesses and suffer from several attacks such as Secret parameter reveal, traceability attack and backward traceability attack. To this aim, we start with describing the process of *ISLAP* protocol firstly.

A. *ISLAP Protocol*

Recently, *ISLAP* proposed a server-less RFID authentication protocol for RFID systems [24] and is a lightweight which makes it suitable for low-cost RFID systems. The structure of *ISLAP* protocol is shown in Fig. 2. Also, notations that are used in the protocol are illustrated in Table II.

TABLE II. THE NOTATIONS OF ISLAP PROTOCOL

Notation	Description
$\mathbf{Rand_i}$	Random number generated by reader R_i
$\mathbf{Rand_j}$	Random number generated by tag T_j
$\mathbf{r_i}$	Identifier of reader R_i
$\mathbf{t_j}$	Secret value of tag T_j
$\mathbf{n_i}$	Message generated by reader R_i
$\mathbf{n_j}$	Message generated by tag T_j
$\mathbf{Seed_{TJ}}$	The secret value shared between R_i and T_j
$\mathbf{Seed_{PTJ}}$	The previous secret value stored in reader R_i
$\mathbf{M(.)}$	Hash function
$\mathbf{A \oplus B}$	Message A is XORed with message B
‖	Concatenation operation

The structure of *ISLAP* protocol that is shown in Fig. 2 is summarized as follows,

a) Authentication phase

Step 1. Reader → Tag: The reader generates $Rand_i$ as a random number and sends it to the tag.

Step 2. Tag → Reader: the tag generates random number $Rand_j$ and computes $n_j = P(Seed_T \oplus (rand_i \parallel rand_j))$ then sends $Rand_j$ and n_j to the reader.

Step 3. Reader → Tag: The reader calculates the following messages:

```
$For\ each\ Seed_{PT_j}\ and\ Seed_{T_j}$ in DB
  $Verify\ n_j \overset{?}{=} P(Seed_X \oplus (rand_i \parallel rand_j))$
  $If\ \ P(Seed_{T_j} \oplus (rand_i \parallel rand_j)) = n_j$
     $X = T_j$
  $Else$:
     $P(Seed_{PT_j} \oplus (rand_i \parallel rand_j)) = n_j$
       $X = PT_j$
```

Then computes $S = M(Seed_X)$, $n_i = P(S)$ and updates $Seed_{PT_j} = Seed_{T_j}$ and $Seed_{T_j} = M(Seed_j)$. Afterwards, the reader forwards the message n_i to the tag.

Step 4. Tag: Once receiving the message n_i, the tag uses its keys to verifies $n_i \overset{?}{=} P\left(M(Seed_{Tj})\right)$. If the verification passed, the tag updates $Seed_{Tj} = M\left(M(Seed_{Tj})\right)$ for another session, otherwise, the tag terminates the protocol.

A. *Secret Parameter Reveal*

In this subsection, we show which an attacker how can reveal secret parameter$Seed_T$. This attack consists of two phases as follows.

Learning phase: In this phase, the attacker is as an eavesdropper. After one successful run, he/she saves the exchanged data between the target tag and the reader including n_i that the reader it sent to the tag.

Attack phase: Then, the attacker uses $n_i = P(M(S)) = P\left(M(Seed_{T,j})\right)$ which it is the obtained data in the learning phase and performs the following steps,

a) Since $M(Seed_{T,j})$ is a L-bit string, thus $M(Seed_{T,j}) \epsilon\, U$ where $U = \{U_1, U_2, \dots, U_{2^l}\}$. Now,

```
            $For\ 1 \le t \le 2^l$
   $Choose\ u_t \in U$
   $if\ \ n_i = P(u_t)\ \ then$
     $return\ u_t\ as\ M(Seed_{T,j})$
$End$
```

Now, it can be seen that the value $M(Seed_{T,j})$ will be used to compute the secret value of the tag in the next session. As result, the attacker can obtain the secret value $Seed_{T,j+1} = P\left(M(Seed_{T,j})\right)$.

In order to perform this attack, the attacker needs to eavesdrop the transmitted data between the reader and the tag in a round and also needs 2^l *PRNG* computations. Noticed that, by betting secret values of the tag, the attacker can perform almost any possible attack including tag impersonation, reader impersonation, traceability attacks and even de-synchronization attack with the success probability of “1”.

B. *Traceability Attack*

One of the main drawback of *ISLAP* protocol is the structure of n_j which lead the protocol to traceability attack in the tag. This attack is summarized as follows.

Learning phase: In round (j), the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, j)$ by sending $rand_i^{T_0}$and obtains $n_j^{T_0}$and $rand_j^{T_0}$. As a result the tag does not updating secret values. Since the length of $Seed_T$ is L-bit, thus $Seed_T\ \epsilon\ Z$ where $Z = \{z_1, z_2, \dots, z_{2^l}\}$. it calculates $Seed_{T,i}^{T_0}$ as follows,

```
$For\ 1 \le q \le 2^l$
          $Choose\ z_q \in Z$
  $if\ \ n_j^{T_0} = PRNG(z_q \oplus (rand_i^{T_0} \parallel rand_j^{T_0}))\ then$
```

$$return\ z_q\ as\ Seed_{T,j}^{T_0}$$

End

Note that, $via\ Seed_{T,j}^{T_0}$, the attacker can calculates the secret value of the target tag T_0 in every round such as round n, by n times apply P function and M function on the secret value $Seed_{T,j}^{T_0}$.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(\ T_0, T_1, j)$. According to the randomly chosen bit $b \in \{0,1\}$, the attacker is given a tag $T_b \in \{T_0, T_1\}$. After that, in round $(j+1)$th, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, j+1)$ by sending $rand_i^{T_0}$ and obtains $n_{j+1}^{T_b}$and $rand_{j+1}^{T_b}$. Now the attacker can compute $\zeta = PRNG(Seed_{T,j}^{T_0} \oplus (rand_i^{T_0} \parallel rand_{j+1}^{T_b}))$ at the session $i+1$.

Guess phase: The attacker $\mathcal{A}$ stops the game G, and outputs a bit $b' \in \{0,1\}$ as a guess of bit b. In order to guess b',Then, the attacker $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as a guess of bit b using the following rule.

$$b' = \begin{cases} 0 & if\ \ n_{j+1}^{T_b} = \zeta \\ 1 & otherwise \end{cases}$$

As a result, it can be written that,

$$Adv_A^{upriv}(K) = \ |pr(b' = b) - pr(random\ coin\ flip)| = \left|pr(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \varepsilon.$$

Proof: Since the value of $Seed_T$ is fixed in all rounds, thus $Seed_{T,j}^{T_0} = Seed_{T,j+1}^{T_0}$. Using this fact, if the tag T_b be equal to target tag T_0, the message $n_{j+1}^{T_b}$ is equal ζ.

C. Forward Traceability Attack

In this section, it is shown that *SRMAP* protocol also does not resists against the forward traceability. Due to computing $Seed_t$ in structure of *SRMAP* protocol, if attacker can find $Seed_t$, she is able to calculate $Seed_t$ in next sessions on her own. This attack is summarized as follows.

Learning phase: In the jth round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $Seed_{T,j}^{T_0}$ from tag T_0. Now the attacker can compute $Seed_{T,j+2}^{T_0}$ at the session $j+2$ by two times repeating $PRNG$ of and operator M.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(\ T_0, T_1, j+2)$. According to the randomly chosen bit $b \in \{0,1\}$, the attacker is given a tag $T_b \in \{T_0, T_1\}$. After that, in round $(j+2)$th, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, j+2)$ by sending $Rand_i$ (i.e., the same value as for session i and obtains $\left(n_{j+2}^{T_b}, rand_{j+2}^{T_b}\right)$.

Guess phase: The attacker $\mathcal{A}$ stops the game G, and outputs a bit $b' \in \{0,1\}$ as a guess of bit b. In order to guess b', firstly the attacker $\mathcal{A}$ computes $\alpha = Seed_{j+2}^{T_0} = P\left(M\left(P\left(M\left(Seed_{T,j}^{T_0}\right)\right)\right)\right)$ and $\beta = PRNG(\alpha \oplus (rand_i^{T_0} \parallel rand_{j+2}^{T_0}))$. Then, outputs a bit $b' \in \{0,1\}$ as a guess of bit b using the following rule.

$$b' = \begin{cases} 0 & if\ \beta = n_{j+2}^{T_b} \\ 1 & otherwise \end{cases}$$

Reader		**Tag**
$(Seed_{T_j}, Seed_{PT_j})$		$\left(Seed_{T_j}\right)$
$For\ each\ Seed_{PT_j}\ and\ Seed_{T_j}$in DB $Verify\ n_j \overset{?}{=} P(Seed_X \oplus (rand_i \parallel rand_j))$ $If\ \ P(Seed_{T_j} \oplus (rand_i \parallel rand_j)) = n_j$ $X = T_j$ Else: $P(Seed_{PT_j} \oplus (rand_i \parallel rand_j)) = n_j$ $X = PT_j$ End Computes: $S = \ M(Seed_X)$ $n_i = \ P(S)$ $If\ \ X = T_j$ $Seed_{PT_j} \leftarrow Seed_{T_j} \leftarrow M(Seed_j)$ $Else$ if $X = PT_j$ Do nothing End If	$rand_i \rightarrow$ $\leftarrow (n_j,\ rand_j)$ $n_i \rightarrow$	Generates random numbers $rand_j$ $n_j = P(Seed_{T_j} \oplus (rand_i \parallel rand_j))$ $Let\ \ k = M\left(Seed_{T_j}\right)$ $Let\ \ a = P(k)$ $if\ n_j = \ a$ $Seed_{T_j} \leftarrow M(Seed_j)$ Else The reader is not authorized End

Fig. 2. *ISLAP* protocol [24].

As a result, it can be written:

$$Adv_A^{upriv}(k) = \ |pr(b' = b) - pr(random\ coin\ flip)|$$

$$= \left|pr(b' = b) - \tfrac{1}{2}\right| = \left|1 - \tfrac{1}{2}\right| = \tfrac{1}{2} \gg \varepsilon$$

Proof: Since the value of $Seed_T$ is fixed in all rounds, thus $Seed_{T,j}^{T_0} = Seed_{T,j+2}^{T_0}$. Using this fact, the following equations can be written.

$If\ \ T_b = T_0$

$$\beta = PRNG(\alpha \oplus (rand_i^{T_0} \parallel rand_{j+2}^{T_b}))$$

with substituting the value $\alpha = P\left(M\left(P\left(M\left(Seed_{T,j}^{T_0}\right)\right)\right)\right)$, Eq. (1) can be rewritten as follows,

$$= PRNG(P\left(M\left(P\left(M\left(Seed_{T,j}^{T_0}\right)\right)\right)\right) \oplus (rand_i^{T_0} \parallel rand_{j+2}^{T_b}))$$

As we know $Seed_{j+2}^{T_0} = P\left(M\left(P\left(M\left(Seed_{T,j}^{T_0}\right)\right)\right)\right)$, it can be written as:

$$= PRNG(Seed_{j+2}^{T_0} \oplus (rand_i^{T_0} \parallel rand_{j+2}^{b}))$$

Finally, with substituting the values of $T_b = T_0$,as result,

$$= PRNG(Seed_{j+2}^{T_b} \oplus (rand_i^{T_0} \parallel rand_{j+2}^{T_b}))$$

$$= n_{j+2}^{T_b}$$

5 Improvement over the Two Protocols

It is shown which both the protocols of SRMAP and ISLAP. have some drawbacks and cannot provide a secure protocol for RFID systems. In order to remove all the reported weaknesses on SRMAP protocol and *ISLAP* protocol, we propose some improvements on their structures and propose two modified version of them.

A. Improved version of SRMAP Protocol

It is observed that in the protocol of *SRMAP* protocol there are two major problems in updating secret keys and structure of I_j that make the protocol vulnerable to traceability attack. In order to solve these problems, we change structure of protocol and the procedure of updating I_j as follows,

$$I_{j+1} \leftarrow PRNG\left(Ack_j \oplus R_t\right) \oplus PRNG\left(I_j\right)$$

$$I_{jT} = I_j \oplus R_t$$

Then, in order to prevent of forward traceability attack we apply some changes in the tag's response and we define a new message N_1 as follows,

$$M_1 = PRNG\left(EPC_s \oplus Ack_j \oplus R_r\right)$$

$$N_1 = PRNG(Auk_j\ ) \oplus PRNG(R_t)$$

Then we change updating procedure,

$$Ack_{i+1} \leftarrow PRNG\left(Ack_j \oplus R_t\right)$$

$$Auk_{i+1} \leftarrow PRNG\left(Auk_j \oplus R_t\right)$$

And in order to prevent of forward secrecy attack which proposed in [14], we change the structure of message $A = H(Rid_k \oplus R_r \oplus N_1)$ to $A = H(Rid_k \oplus R_r \oplus N_1)$.

Note that, in the improved version, in order to increasing the security and privacy of the protocol the tag does not send the random number R_t to the reader which it is described with more detail in the rest of paper. The improved version of *SRMAP* protocol described as follows, also it can be summarized in Fig. 3

a) Initial phase

This phase is the same of *SRMAP* protocol.

b) Authentication phase

This phase includes six steps as follows,

Step 1. Reader → Tag: The reader generates R_r as a random number and sends it to the tag.

Step 2. Tag → Reader: Once receiving R_r, the tag generates a random number R_t. It computes the following messages then sends M_1, N_1 and I_{jT} to the reader.

$$M_1 = PRNG\left(EPC_s \oplus Ack_j \oplus R_r\right)$$

$$N_1 = PRNG(Auk_j\ ) \oplus PRNG(R_t)$$

$$I_{jT} = I_j \oplus R_t$$

Step 3. Reader → Server: The reader calculates $A = H(Rid_k \oplus R_r \oplus M_1)$ and forwards the messages $(M_1, I_{jT}, N_1, A, R_r)$ to the server.

Step 4. Server → Reader: Based on the received messages from the reader, the server performs the following operations,

a) The server verifies $H(Rid_k \oplus R_r \oplus N_1) \overset{?}{=} A$ with R_r and follows the rest of authentication procedure.

b) The back-end server first computes $R_{t,X} = I_{\mathrm{jT}} \oplus I_X$ for $X \in \{old, new\}$. Then it checks whether $PRNG(EPC_s \oplus Ack_x \oplus R_r) \overset{?}{=} M_1$ and determines that $X = old$ or new and it authenticates the tag if $PRNG(EPC_s \oplus Ack_x \oplus R_r)\ _{=} M_1$ for one of the values of X .Then it verifies $N_1 \overset{?}{=} PRNG(\ Auk_x) \oplus PRNG(R_{t,x})$ to authenticate the tag. The server responds to the reader by the following messages,

$$M_2 = \ PRNG(EPC_s \oplus R_t) \oplus Ack_j$$

$$B = data_i \oplus Rid_k$$
$$C = H(data_i \oplus R_r)$$

otherwise, the back-end server aborts the protocol.

c) Finally, the server updates its secret values as follows,

```
$If\ \ X = new$
    $Ack_{old} \leftarrow Ack_{new} \leftarrow PRNG(Ack_j \oplus R_{t,X})$
    $Auk_{old} \leftarrow Auk_{new} \leftarrow PRNG(Auk_j \oplus R_{t,X})$
    $I_{old} \leftarrow I_{new} \leftarrow PRNG(Ack_j \oplus R_{t,X}) \oplus PRNG(I_j)$
$Else$
    $Do\ nothing$
$End$
```

Step 5. Reader → Tag: Now using the received message B, the reader computes $data_i = B \oplus Rid_k$, and verifies $H(data_i \oplus R_r) \overset{?}{=} C$. If the verification is successful, the reader sends M_2 to the tag.

Step 6. Utilizing the received message M_2, the tag uses its keys to verifies $M_2 \overset{?}{=} PRNG(Auk_j \oplus R_t) \oplus Ack_j$. If the verification passed, the tag computes $Ack_{j+1} \leftarrow PRNG(Ack_j \oplus R_t)$, $Auk_{j+1} \leftarrow PRNG(Auk_j \oplus R_t)$ and $I_{j+1} \leftarrow PRNG(Ack_j \oplus R_t) \oplus PRNG(I_j)$. Eventually, the tag replaces Auk_j, Ack_j and I_j with Auk_{j+1}, Ack_{j+1} and I_{j+1} respectively, for another session, otherwise, the tag aborts the protocol.

B. Improved version of ISLAP Protocol

In Section 4, we have seen that *ISLAP* protocol has some problems that make it vulnerable against both security and privacy attacks. In the proposed protocol, in order to enhance the security and the privacy and remove all mentioned attacks, we apply some changes in the updating procedures and tag response. More precisely, in order to prevent secret reveal attack, we have changed tag response as follows,

$$n_j = M(Seed_{T_j} \oplus (rand_i \parallel rand_j))$$

Where $M(.)$ is a one way hash function.

Also it is observed that in the protocol of *ISLAP* protocol is vulnerable to traceability and forward traceability attacks. In order to remove these weaknesses, we change the procedure of updating in the improved version as follows,

$$Seed_{T_j} \leftarrow M(Seed_j \oplus rand_j)$$

Database	**Reader**	**Tag**
$(Ack_{old}, Auk_{old}, I_{old}, Ack_{new}, Auk_{new}, I_{new}, Rid_k, EPC, data)$	Rid_k	$(Ack_i, Auk_i, I_i, EPC_s)$
$For\ each\ Rid_k$in DB $Verify\ H(Rid_k \oplus R_r \oplus M_1) \overset{?}{=} A$ $Computes\ \ R_{t,new} = I_{new} \oplus I_{jT}$ and $R_{t,old} = I_{old} \oplus I_{jT}$ $Verify\ \ PRNG(EPC_s \oplus Ack_x \oplus R_r) \overset{?}{=} M_1$ $If\ \ PRNG(EPC_s \oplus Ack_{new} \oplus R_r) \overset{?}{=} M_1$ $X = new$ Else: $PRNG(EPC_s \oplus Ack_{old} \oplus R_r) \overset{?}{=} M_1$ $X = old$ End $Verify\ \ N_1 \overset{?}{=} PRNG(\ Auk_x) \oplus PRNG(R_{t,x})$	$R_r \rightarrow$ $\leftarrow (M_1, I_{jT}\ , N_1,)$ $A = H(Rid_k \oplus R_r \oplus N_1)$ $\leftarrow (M_1, I_{jT}\ , R_r, A, N_1)$	Generates R_t $M_1 = Auk_j \oplus PRNG(EPC_s \oplus Ack_j \oplus R_r) \oplus PRNG(R_t)$ $N_1 = PRNG(Auk_j\ ) \oplus PRNG(R_t)$ $I_{jT} = I_j \oplus R_t$
Computes: $M_2 = \ PRNG(EPC_s \oplus R_t) \oplus Ack_j$ $B = data_i \oplus Rid_k$ $C = H(data_i \oplus R_r)$ $If\ \ X = new$ $Ack_{old} \leftarrow Ack_{new} \leftarrow PRNG(Ack_j \oplus R_{t,X})$ $Auk_{old} \leftarrow Auk_{new} \leftarrow PRNG(Auk_j \oplus R_{t,X})$ $I_{old} \leftarrow I_{new} \leftarrow PRNG(Ack_j \oplus R_{t,X}) \oplus PRNG(I_j)$ $Else$ Do nothing End If	$(M_2, B\ , C) \rightarrow$ $data_i = B \oplus Rid_k$ $Verify\ \ H(data_i \oplus R_r) \overset{?}{=} C$ $M_2 \rightarrow$	$Verify\ M_2 \overset{?}{=} PRNG(EPC_s \oplus R_t) \oplus Ack_j$ $Ack_{i+1} \leftarrow PRNG(Ack_j \oplus R_t)$ $Auk_{i+1} \leftarrow PRNG(Auk_j \oplus R_t)$ $I_{i+1} \leftarrow PRNG(Ack_j \oplus R_t) \oplus PRNG(I_i)$

Fig. 3.The Improved Version of SRMAP protocol.

And also we have change reader response as follows,

$$n_i = M(S \oplus rand_i)$$

The improved version of *ISLAP* protocol shown in Fig. 4 and it described with more detail as follows,

c) *Initial phase*

This phase is the same of *SRMAP* protocol.

d) *Authentication phase*

This phase includes four steps as follows,

Step 1. Reader → Tag: The reader generates $rand_i$ as a random number and sends it to the tag.

Step 2. Tag → Reader: Once receiving $rand_j$, the tag generates a random number $rand_j$. It computes the following message then sends n_j and $rand_j$ to the reader.

$$n_j = M(Seed_{T_j} \oplus (rand_i \parallel rand_j))$$

Step 3. Reader → Tag: The server performs the following operations,

a) The reader first select $Seed_X$ for $X \in \{PT_i, T_i\}$. Then it checks whether $n_j \overset{?}{=} H(Seed_X \oplus (rand_i \parallel rand_j))$ and determines that $X = PT_i, T_i$ and it authenticates the tag if $n_{j\,=} H(Seed_X \oplus (rand_i \parallel rand_j))$ for one of the values of X .Then it responds to the tag by the following messages,

$$S = M(Seed_X)$$

$$n_i = M(S \oplus rand_i)$$

b) At the end, the reader updates its secret values as follows,

```
$If\ \ X = T_i$
    $Seed_{PT_j} \leftarrow Seed_{T_j} \leftarrow M(Seed_j \oplus rand_j)$
$If\ X = PT_i$
    $Do\ nothing$
$End$
```

Step 5. The tag uses its keys to comp*ute* $k = M(Seed_{T_j})$ *and* $a = M(k \oplus rand_i)$ *then it* verifies $n_j \overset{?}{=} a$. *If the verification* passed, the tag authenticates the reader and updates its secret key as follows,

$$a = M(k \oplus rand_i)$$

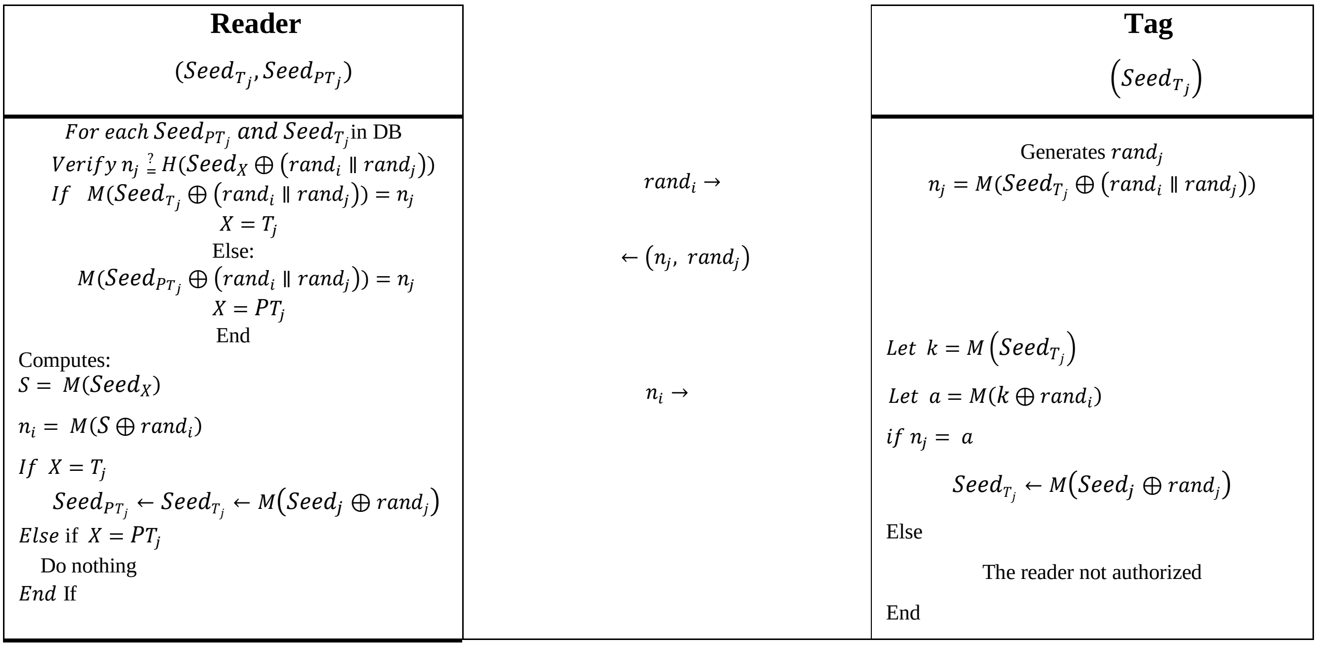
otherwise, the tag aborts the protocol.

In Table II, the comparison of the security and the privacy of the improved protocols with some other protocols that are in the same class are summarized. As it can be seen, the two proposed protocols are secure against various attacks and can protect RFID users against different threats.

TABLE III. ANALYSES OF THE PROPOSED PROTOCOLS

Properties	**Impersonation**	**Secret Parameter Reveal**	**Traceability**	**Forward Traceability**
***SRMAP* [21]**	×	×	×	×
Improved *SRMAP*	⊙	⊙	⊙	⊙
Deng [22]	×	×	×	⊙
ISLAP [24]	⊙	×	×	×
Improved ISLAP	⊙	⊙	⊙	⊙

⊙: Secure × Insecure



Reader

$(Seed_{T_j}, Seed_{PT_j})$

```
$For\ each\ Seed_{PT_j}\ and\ Seed_{T_j}$ in DB
$Verify\ n_j \overset{?}{=} H(Seed_X \oplus (rand_i \parallel rand_j))$
$If\ \ M(Seed_{T_j} \oplus (rand_i \parallel rand_j)) = n_j$
    $X = T_j$
Else:
    $M(Seed_{PT_j} \oplus (rand_i \parallel rand_j)) = n_j$
    $X = PT_j$
End
Computes:
$S = M(Seed_X)$
$n_i = M(S \oplus rand_i)$
$If\ \ X = T_j$
    $Seed_{PT_j} \leftarrow Seed_{T_j} \leftarrow M(Seed_j \oplus rand_j)$
$Else$ if $X = PT_j$
    Do nothing
$End$ If
```

$rand_i \rightarrow$

$\leftarrow (n_j,\ rand_j)$

$n_i \rightarrow$

Tag

$(Seed_{T_j})$

```
Generates $rand_j$
$n_j = M(Seed_{T_j} \oplus (rand_i \parallel rand_j))$
$Let\ \ k = M(Seed_{T_j})$
$Let\ \ a = M(k \oplus rand_i)$
$if\ n_j = a$
    $Seed_{T_j} \leftarrow M(Seed_j \oplus rand_j)$
Else
    The reader not authorized
End
```

Fig. 4.The proposed improved version of ISLAP protocol.

6 Conclusion

We have analyzed two RFID authentication protocols in terms of security and privacy which are proposed by *SRMAP* and ISLAP. It is shown both protocols have some drawbacks and are not secure against various attacks. We have shown that *SRMAP* protocol cannot resists traceability attacks such as *traceability* and *forward traceability* attacks. Moreover, we have shown that *ISLAP* protocol suffer from security and privacy attacks and cannot safeguard RFID users and is vulnerable against *Secret parameter reveal*, *traceability* and *forward traceability* attacks. Furthermore, an improved version of each motioned protocol is proposed. Our analyses show that The improved protocols can resist all of the mentioned weaknesses and are able to prevent those attacks as we have shown in the analyses table.

References

[1] D. Heyden, "RFID Applications," 6 2016. [Online]. Available: http://www.fibre2fashion.com/industry-article/3271/rfid-applications#sthash.zRakWiB9.

[2] K. Mahajan, P. Pandey, B. K. Pandher, "Application of RFID Technology in Libraries and Role of Librarian," 2010.

[3] C. Oberli, M. Torres-Torriti, D. Landau, "Performance evaluation of UHF RFID technologies for real-time passenger recognition in intelligent public transportation systems," *IEEE transactions on intelligent transportation systems*, vol. 11, no. 3, pp. 748-753, 2010.

[4] S. M. Alavi, K. Baghery and B. Abdolmaleki, "Traceability analysis of recent RFID authentication protocols," *Wireless Personal Communications*, vol. 83, no. 3, pp. 1663-1682, 2015.

[5] S. Vaudenay, "On privacy models for RFID," in *ASIACRYPT 2007, LNCS 4833, pp. 68–87.*, 2007.

[6] G. Avoine, "Adversarial model for radio frequency identification," *Cryptology ePrint Archive, report 2005/049. http://eprint.iacr.org/2005/049,* 2005.

[7] C. H. Lim, and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer," *In Proceedings of ICICS '06, LNCS 4307 ,* pp. 1-20, 2006.

[8] K. Ouafi and R. C.-W. Phan, "Privacy of recent RFID authentication protocols," in *4th International Conference on Information Security Practice and Experience (ISPEC), Springer*, 2008.

[9] A. Juels, and S.A Weis, "Defining strong privacy for RFID," in *Proceedings of PerCom '07, pp. 342–347.* , 2006.

[10] B. Alomair, A. Clark, J. Cuellar, R. Poovendran, "Scalable RFID systems: a privacy-preserving protocol with constant-time identification," *IEEE Transactions on Parallel and Distributed Systems,* vol. 23, no. 8, pp. 1536-1550, 2012.

[11] K. Baghery, B. Abdolmaleki, B. Akhbari and M. R. Aref, " Enhancing Privacy of Recent Authentication Schemes for Low-Cost RFID Systems," *The ISC International Journal of Information Security,* vol. 7, no. 2, pp. 135-149, 2016.

[12] E.-J. Yoon, "Improvement of the securing rfid systems conforming to EPC class 1 generation 2 standard," *Expert Syst. Appl.*, vol. 39, no. 11, p. 1589–1594, 2012.

[13] M.H. Habibi, M. R. Alaghband, and M. R. Aref, "Attacks on a lightweight mutual authentication protocol under EPC C-1 G-2 standard," in *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, Springer, 2011, pp. 254-263.

[14] B. Abdolmaleki, K. Baghery, B. Akhbari, M. Aref, "Cryptanalysis of two EPC-based RFID security schemes," in *Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on*, 2015.

[15] N. Bagheri, M. Safkhani and M. Naderi, "Cryptanalysis of a new EPC class-1 generation-2 standard compliant RFID protocol," *Neural Computing and Applications,* vol. 24, no. 3-4, pp. 799-805, 2014.

[16] S. M. Alavi, B. Abdolmaleki, K. Baghery, "Vulnerabilities and improvements on HRAP+, a hashbased RFID authentication protocol," *Advances in Computer Science: an International Journal*, vol. 3, no. 6, pp. 51-56, 2014.

[17] Z. Liu, D. Liu, L. Li, H. Lin and Z. Yong, " Implementation of a new RFID authentication protocol for EPC Gen2 standard," *IEEE Sensors Journal,* vol. 15, no. 2, pp. 1003-1011, 2015.

[18] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador and A. Ribagorda, "Lightweight cryptography for low-cost RFID tags," 2016, pp. 121-150.

[19] C. Jin, C. Xu, X. Zhang and F. Li, " A Secure ECC-Based RFID Mutual Authentication Protocol to Enhance Patient Medication Safety," *Journal of medical systems,* vol. 40, no. 1, pp. 1-6, 2016.

[20] J. Cho, Y. Jeong, S. O. Park, "Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol," *Computers & Mathematics with Applications,* vol. 69, no. 1, pp. 58-65, 2015.

[21] C. C. Chang, W. Y. Chen, T. F. Cheng, "A Secure RFID Mutual Authentication Protocol Conforming to EPC Class 1 Generation 2 Standard," in

Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on, 2014.

[22] M. Deng, W. Yang, W. Zhu, "Weakness in a serverless authentication protocol for radio frequency identification," in *Mechatronics and Automatic Control Systems*, Springer, 2014, pp. 1055-1061.

[23] S. Abdolmaleky, S. Atapoor, M. Hajighasemlou, H. Sharini, "A Strengthened Version of a Hash-based RFID Server-less Security Scheme," *Advances in Computer Science: an International Journal,* vol. 4, no. 3, pp. 18-23, 2015.

[24] M. Pourpouneh, R. Ramezanian, F. Salahi, "An Improvement over a Server-less RFID Authentication Protocol," 2015.