Investigating the Performances and Vulnerabilities of Two New Protocols Based on R-RAPSE

Seyed Salman Sajjadi Ghaemmaghami¹, Afrooz Haghbin² and Mahtab Mirmohseni³

¹ Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran
Seyed.sajjadi@srbiau.ac.ir

² Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran
Afrooz.haghbin@srbiau.ac.ir

³ Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
mirmohseni@sharif.edu

Abstract
Radio Frequency IDentification (RFID) is a pioneer technology which has depicted a new lifestyle for humanity in all around the world. Every day we observe an increase in the scope of RFID applications and no one cannot withdraw its numerous usage around him/herself. An important issue which should be considered is providing privacy and security requirements of an RFID system. Recently in 2014, Cai et al. proposed two improved RFID authentication protocols based on R-RAPS rules by the names of IHRMA and I2SRS. In this paper, we investigate the privacy of the aforementioned protocols based on Ouafi and Phan formal privacy model and show that both IHRMA and I2SRS protocols cannot provide private authentication for RFID users. Moreover, we show that these protocols are vulnerable to impersonation, DoS and traceability attacks. Then, by considering the drawbacks of the studied protocols and implementation of messages with new structures, we present two improved efficient and secure authentication protocols to ameliorate the performance of Cai et al.’s schemes. Our analysis illustrate that the existing weaknesses of the discussed protocols are eliminated in our proposed protocols.

Keywords: Authentication, RFID protocol, Privacy, Security, Ouafi Phan privacy model, Traceability, Impersonation

1. Introduction
Nowadays, our world is transitioning from an internet of connected individuals to an internet in which everything and everyone is connected, also known as Internet of Things (IoT) [1]. Radio Frequency IDentification (RFID) is a technology which provides a contactless identification through magnetic waves. Health-care, livestock and animal tracking, access control, transportation and supply chain can be mentioned as its applications which play an important roles to prepare the structures for developing the concept of IoT [2-6]. As it is shown in Fig. 1, RFID systems involve three main parts: back-end server, reader and tag. The tag is a microchips which can be attached to different objects with different purposes in an RFID system that falls in one of the three classes: active, passive and semi-active [7]. A passive tag does not have any battery and obtains sufficient energy to reply the reader from the magnetic field achieved through sending the request by the reader. An active tag contains an inner battery, allows it to start a new connection with the reader over than only be a responder. Although the semi-active tag holds an inner battery, it just responds to the received queries from the reader, and performing the internal operations are the only usage of the internal battery [8]. Decreasing the size and cost of RFID tags, have been led to popularity and vast implementation of passive tags in most of novel applications. The back-end server stores all the information of the tags and the readers, and establishes a connection with the tag via tranceiving data with the reader and after investigating the correctness of transferred messages, authenticates the reader and the tag. Although, RFID technology is developing rapidly and providing comfort for users, deficiency of supplying the necessary security, will result in irreparable damages [9]. Therefore, scholars have proposed various type of protocols to provide security and privacy of end-users in RFID systems, which generally classify into four classes based on the deployed cryptographic functions [10]. Full-fledged are the first classes, include ordinary cryptographic functions such as public or private key cryptography systems, one-way hash functions and so forth [2]. Random Number Generators (RNG) and one-way
hash functions are permitted to use in the second class. The third class is called lightweight, includes RNG functions and Cyclic Redundancy Code (CRC) checksums[4]. Finally, ultra-lightweight is the last classification, limited to the usage of simple bitwise operators such as AND, OR and XOR [11].

By paying attention to the mentioned classification, several protocols have been presented in the last few years [6, 12-16]. Yeh et al. proposed an RFID authentication protocol based on EPC Class 1 Generation 2 standard in 2010 which supplies tag privacy [6]. In 2011, Yoon declared that Yeh et al.’s protocol is still vulnerable to data integrity and forward secrecy problems [13]. So, he suggested an improved protocol available in places with high level of security. In 2011, Cho et al. proposed a hash-based RFID mutual authentication protocol [17]. They believe that their scheme solves the privacy and forgery problems as well as provides all the security requirements of RFID users [17]. In 2014, Cai et al. investigated Cho et al. and Yoon’s protocols through their paper [18]. They believe that preparing the authentication procedure should be designed and confirmed theoretically before testing experimentally. So, they defined rules by the name of R-RAPSE which provides security and privacy for RFID protocols [18]. They found that Cho et al.’s protocol does not provide data integrity. Moreover, it is vulnerable to de-synchronization attack, therefore they proposed the IHRMA protocol based on R-RAPSE rules, which is an improved version of Cho et al.’s protocol. They also diagnosed that Yoon’s protocol cannot preserve the location of tag owner, which results in weaknesses for providing privacy issues. Thus, they proposed the I2SRS protocol as a modified for Yoon’s protocol [7].

In this paper, we analyze the IHRMA and I2SRS protocols and show that there are still some flaws in their protocols. It is exposed that similarity between the generated messages and updating procedures will make the IHRMA protocol vulnerable to tag and reader impersonation. Besides revealing the secret key in the I2SRS protocol, shown that an attacker can perform tag impersonation, traceability and forward traceability attacks after maximum $2^{16}$ computations and backward traceability attack with $2^{17}$ runs in the worst case.

Recently, different types of privacy model have been proposed to study the authentication routine in RFID protocols [19-26]. Among these formal privacy models, Ouafi-Phan privacy model [23] is one of the well-known privacy models which has been proposed in 2008 and due to pertinent queries for different privacy analysis, it has got more attention by researchers [18-22]. We use this privacy model for privacy analysis of the IHRMA and I2SRS protocols. A summary of Ouafi-Phan privacy model and its usage in privacy analysis can be found in our last paper [7].

The rest of this paper is organized as follows: In Section 2 we analyze the IHRMA protocol and its vulnerabilities are discussed. We do the same for the I2SRS protocol in Section 3. Our improved RFID authentication protocols are presented in Section 4. The proposed protocol are compared with some existing ones in the terms of security and privacy in Section 5. Finally, we conclude the paper in Section 6.

2. IHRMA protocol

2.1 Analyzes of IHRMA protocol

Cai et al. believe that R-RAPSE rules provide sufficient security in their protocol [18]. So, to prevent de-synchronization of secret information in two sides of protocol, they proposed their improved protocol based on R-RAPSE instructions which provide data integrity via hash functions in their method. The structure of IHRMA protocol is depicted in Fig. 2. The connections between the reader and the back-end server are secure, while the communications between the reader and the tag are insecure. In this section, we analyze the IHRMA protocol and prove that their protocol is vulnerable to tag and reader impersonation attacks. The notations used in this protocol are as follow:

- $RID_i$: Group ID of random number.
- $ID_k$: The ID of the tag k.
- $S_j$: Secret value mutually shared between the server and the tag, used in the jth session.
- $R_t$: Random number generated by the tag.
- $R_r$: Random number generated by the reader.
- $C_1$: Data generated by the tag for authentication.
- $C_2$: Blind factor.
2.2 Tag impersonation attack

Here now we show that the IHRMA protocol cannot prevent an adversary \( \mathcal{A} \) from performing impersonation attacks. Cai et al. have used mod operator in construction of \( RID_i \) and they have believed that it will result better performance of their protocol [18]. \( RID_i \) is described as follow:

\[
RID_i = (R_t - R_t \mod S_j + 1) (0: 47) \parallel (R_t + S_j - R_t \mod S_j) (48: 95)
\]

**Notion1**: There is a characteristic in the mod operator, which is described here. If we have \( a \) and \( b \) as two integers and \( a < b \), then \( a \mod b \) will result in \( a \).

Using Notion 1, in the IHRMA protocol [18], there will be a situation where \( R_t < S_j \), results in

\[
RID_i = (R_t - R_t + 1) (0: 47) \parallel (R_t + S_j - R_t) (48: 95)
\]

\[
RID_i = (1) (0: 47) \parallel (S_j) (48: 95)
\]

Now, an adversary can perform tag impersonation attack in two phases as follows:

**Learning phase**: The attacker \( \mathcal{A} \) sends an execute query \( (R, T_0, i) \), obtains \( \{R_r, C_1, C_2 \oplus R_t, C_3, C_4, C_5\} \). Then by changing the value of \( R_t \), which makes new messages as \( C_4 \) and \( C_5 \), prevents the back-end server authentication in the tag side. Therefore, the secret value will not be updated in the tag.

**Attack phase**: In session \( (i + 1) \), the attacker impersonates him/herself as a trusted tag. To this aim, the adversary performs as follows:

A. After receiving \( R_t' \) from the reader in the \( (i + 1) \) session, the attacker \( \mathcal{A} \) calculates \( C_{1,\text{Fic}} \) and \( D_{1,\text{Fic}} \) with the values stored in the last session, and sends them to the reader.

\[
C_{1,\text{Fic}} = C_1 \tag{3}
\]

\[
D_{1,\text{Fic}} = C_2 \oplus R_t \oplus R_t \oplus R_t' \tag{4}
\]

B. Reader puts \( R_t'^{+1} \) beside \( C_{1,\text{Fic}} \) and \( D_{1,\text{Fic}} \), sends them all to the back-end server.

C. The back-end server calculates \( C'_2 \) and obtains \( R_t'^{+1} \) as described below:

---

| Back-end Server \((ID_k, S_j, S_{j-1}, DATA)\) | Reader | Tag \((ID_k, S_j)\) |
|-----------------------------------------------|--------|----------------|
| \(\text{1.1 Request: } R_r\) & \(R_r\) | \(\text{2.1 } C_3 = H(ID_k \oplus R_t \oplus R_t \oplus RID_i)\) | \(\text{2.3} \) |
| \(\text{1.2} \rightarrow \) | \(\text{2.2 } C_2 = S_j (0: 47) \parallel ID_k (48: 95)\) | \(\text{2.3} \) |
| \(\text{3.1} \leftarrow C_1, C_2 \oplus R_t, R_r\) | \(\text{5.1} \) | \(\text{6.1} C_3 = H(C_2 \oplus RID_i)\) |
| \(\text{4.9 updating:} \) \(S_j - 1 \leftarrow S_j\), \(S_j \leftarrow S_j + 1\) | \(\text{6.2 updating:} \) \(S_j \leftarrow S_j + 1\) | \(\text{1.2} \rightarrow \) |
By computing $R_t^{i+1}$, the back-end server calculates $RID_{i+1}$. As we stated before, if $R_t < S_j$, the obtained $RID_{i+1}$ is equal with eq. (1).

### D. Learning phase

Moreover, by considering the conformation of equation (2), the tag will generate a new $R_t^{i+1}$, calculate $C_4^{i+1}$ and send it with $D^{i+1}$ to the counterfeet reader which can be written as,

$$C_4^{i+1} = H(ID_t \oplus R_t^{i+1} \oplus R_t^{i+1} \oplus RID_{i+1})$$

$$D^{i+1} = C_2^{i+1} \oplus R_t^{i+1}$$

### E. Eavesdropping phase

The back-end server, computes $C_1^{i(i+1)}$ and authenticates the tag, if $C_1^{i(i+1)} = C_{1,FIC}$. By considering the situation which $R_t < S_j$ and $R_t^{i+1} < S_j$, $RID_{i+1} = RID_i$ and the back-end server will verify the tag as a legal one, update its secret values and send $C_3^{i+1}$, $C_4^{i+1}$ and $C_5^{i+1}$ to the reader. Ultimately, the back-end server endorses the falsified tag as an allowable one.

**Proof:**

$$C_1^{i(i+1)} = H(ID_t \oplus R_t^{i+1} \oplus R_t^{i+1} \oplus RID_{i+1})$$

$$= H(ID_t \oplus R_t^{i+1} \oplus R_t \oplus R_t^{i+1} \oplus RID_{i+1})$$

$$= H(ID_t \oplus R_t \oplus R_t \oplus RID_i)$$

$$= C_{1,FIC}$$

The adversary will succeed, if the assumptions are correct. For random selection of $R_t$ and $R_t^r$, the success probability of each assumption is $\frac{1}{2}$. The total probability of the above attack is $\frac{1}{4}$, while it’s complexity of is two runs of the protocol.

### 2.3 Reader impersonation attack

As discussed in subsection 2.2 the structure of generating $RID_i$ in the IHRMA protocol which is shown in equation (1), allows the attacker to perform tag impersonation attack. Now, we want to express that this type of producing $RID_i$, will result in reader impersonation attack, with the success probability of $\frac{1}{4}$, too. Its implementation can be described as follows:

#### Learning phase: The attacker eavesdrops the $i$th session of the protocol by sending an execute query ($R_t, T_{0}, i$) and obtaining $\{R_t, C_1, C_2 \oplus R_t, C_3, C_4, C_5\}$. Now, to prevent updating secret values, the attacker blocks the 5.1 step of the protocol.

#### Attack phase: The attacker $\mathcal{A}$ acts as a reader and starts a new session by generating $R_t^{i+1}$ randomly and sending it to the tag $T_{0}$, that leads to reader impersonation attack. The manner of these steps are discussed below:

1. The tag did not update its secret values in the last session. Moreover, by considering the conformation of

$$R_t^{i+1} = D_{t,FIC} \oplus C_2$$ (5)
4.1 For each \(ID_r\) in server
   - Verify \(H(ID_r) \oplus R_r \neq V\)
   - For each tuple \((EPC_c; K_o, K_n)\)
     - Calculate \(I_0 = M_1 \oplus K_o\) and \(I_n = M_1 \oplus K_n\)
     - Verify \(I_0 \neq P(EPC_c \oplus R_r \oplus \beta \oplus K_o)\)
     - Verify \(I_n \neq P(EPC_c \oplus R_r \oplus \beta \oplus K_n)\)

4.2 Calculate \(x = 0 / N\)
4.3 Verify \(\beta \oplus K_x \oplus P(\alpha_i \oplus K_x) \neq \gamma\)
4.4 Calculate \(R_x^* = \beta \oplus K_x\)
4.5 Calculate \(M_2 = P(EPC_c \oplus R_x^* \oplus P_x)\)
4.6 Calculate \(Info = DATA \oplus ID_r\)
4.7 \(MAC = H(DATA \oplus R_r)\)

4.8 Updating
   - If \(x = N\)
     - \(K_o \leftarrow K_n \leftarrow P(K_n)\)
     - \(P_o \leftarrow P_n \leftarrow P(P_n)\)
     - \(\alpha_o \leftarrow \alpha_n \leftarrow P(R_t \oplus R_r)\)
   - Else
     - \(\alpha_o \leftarrow P(R_t \oplus R_r)\)
   - End if

\[
\begin{align*}
5.1 \ DATA &= \text{Info} \oplus \text{ID}_r \\
5.2 \ Verify \ H(DATA \oplus R_r) &= MAC \\
5.3 \ M_2 &\rightarrow \text{MAC} \\
6.1 \ Verify \ M_2 \oplus P_i \neq P(EPC_c \oplus R_c) \\
6.2 \ Updating \quad K_{i+1} &= P(K_i) \\
&\quad P_{i+1} = P(P_i) \\
&\quad \alpha_{i+1} = P(R_t \oplus R_r)
\end{align*}
\]

Fig. 3. I2SRS protocol [18].

3.2 Secret parameter Reveal

Presence of RFID technology as a main part of different systems, compels researchers to present novel protocols which provide security and privacy. An important issue to be considered is preventing an adversary to access secret values, but in this section, we will show that the I2SRS protocol is vulnerable to secret parameter reveal which is described below:

**Learning phase:** An attacker eavesdrops one session of the protocol and blocks the step 5.3. Therefore, it obtains \(\{M_1^i, \beta^i, \gamma^i, \alpha_i, R_t^i, V, M_2^i\}\) while the tag did not update its secret values.

**Attack phase:** The attacker uses stored values from the last session and by paying attention to their similarity, conjectures the secret value \(K_i\) with the probability of 1. The attacker computes \(Z\) as below:

\[
Z = \beta^i \oplus \gamma^i \oplus \alpha_i
\]

\[
Z = R_t \oplus K_i \oplus \alpha_i \oplus P(\alpha_i \oplus K_i)
\]

\[
Z = K_i \oplus \alpha_i \oplus P(\alpha_i \oplus K_i)
\]  \hspace{1cm} (12)

\(K_i\) is a 16 bit string, so it is one of the element of \(\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{2^{16}}\}\). The attacker knows the value of \(\alpha_i\) through the last session, so he/she is able to obtain \(K_i\) after maximum \(2^{16}\) runs as below:

For \(2 < i < 2^{16}\)

Choose \(K_i \in \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{2^{16}}\}\)

\[
W = K_i \oplus \alpha_i \oplus P(\alpha_i \oplus K_i)
\]

If \(Z = W\)

Return \(K_i\) as a correct secret key
End

Therefore, similarity in generating messages through I2SRS protocol, makes it vulnerable to protection of secret parameters. Moreover, lack of precision in updating procedure, allows the attacker \(A\) to calculate \(K_i\) in the \((i + x)\) session by performing \(x\) times \(PRNG\) of \(K_i\).

3.3 Tag Impersonation attack

Revealing the secret parameter in the I2SRS protocol, let an adversary \(A\) impersonate a legitimate tag after maximum \(2^{16}\) runs, which proved in Section 3.2. The
method of applying this attack is as follows,

**Learning phase:** An adversary \( \mathcal{A} \) sends an Execute query \( \langle R, T_0, i \rangle \) and obtains \( \{M_1^i, \beta_1^i, \gamma_1^i, \alpha_i, R_r^i \} \). He/she is able to guess \( K_i \) with the probability of 1, as discussed in Section 3.2 and obtain \( R_t \),

\[
R_t^1 = \beta^1 \oplus K_i \quad (14)
\]

**Attack phase:** In the next session, after sending a request and \( R_t^{i+1} \) to the tag by the reader, the attacker \( \mathcal{A} \) introduces him/herself as a legal tag and the following events take place:

1. The attacker \( \mathcal{A} \) generates \( \{M_1^{i+1}, \beta_1^{i+1}, \gamma_1^{i+1}, \alpha_{i+1} \} \) as below and sends them to the reader,

\[
M_1^{i+1} = M_1^i
\]

\[
\beta_1^{i+1} = R_t^1 \oplus R_r^i \oplus R_r^{i+1} \oplus K_i
\]

\[
\alpha_{i+1} = R_t^1 \oplus R_r^i \oplus R_r^{i+1} \oplus \alpha_i'
\]

\[
\gamma_1^{i+1} = R_t^1 \oplus R_r^i \oplus R_r^{i+1} \oplus P(\alpha_{i+1} \oplus K_i) \quad (15)
\]

As we do not know the value of \( EPC_e \), it is not possible to compute \( M_1^{i+1} \) unless we use the same message as \( M_1^i \) and combine other messages as stated above. In the other words, by calculating \( \alpha_i \) as \( P(R_t \oplus R_t) \) and replacing \( R_t^{i+1} \) with \( R_t^1 \oplus R_r^i \oplus R_r^{i+1} \), Messages are prepared.

2. The reader computes \( V = H(ID_r \oplus R_r^{i+1}) \) and sends \( \{M_1^{i+1}, \beta_1^{i+1}, \gamma_1^{i+1}, \alpha_{i+1}, R_r^{i+1}, \gamma_1^{i+1} \} \) to the back-end server.

3. As the reader is legal, the back-end server authenticates it and to confirm the tag, for each tuple \( (EPC; K_o, K_n) \), calculates \( I_x = M_1 \oplus K_x \). Given that the tag did not update its secret values in the last session, \( x \) refers to the old parameters. Then, the back-end server checks if it is equal with \( (EPC \oplus R_r \oplus K_x) \) where,

\[
M_1 \oplus K_i = P(EPC_s \oplus R_r^{i+1} \oplus R_t^{i+1}) \oplus K_i \oplus K_t
\]

\[
= P(EPC_s \oplus R_r^{i+1} \oplus R_t^1 \oplus R_r^i \oplus R_t^{i+1})
\]

\[
= P(EPC_s \oplus R_t^1 \oplus R_r^i)
\]

\[
= (EPC_s \oplus R_r^{i+1} \oplus K_{i+1} \oplus K_t) \quad (16)
\]

4. The back-end server checks the equality of \( \beta^{i+1} \oplus K_i \oplus P(\alpha_{i+1} \oplus K_i) \equiv \gamma_{i+1} \) and obtains,

\[
\beta^{i+1} \oplus K_i \oplus P(\alpha_{i+1} \oplus K_i)
\]

\[
= R_t^1 \oplus R_r^i \oplus R_r^{i+1} \oplus K_i \oplus K_t \oplus P(\alpha_{i+1} \oplus K_i)
\]

\[
= R_t^1 \oplus R_r^i \oplus R_r^{i+1} \oplus P(\alpha_{i+1} \oplus K_i) = \gamma_{i+1} \quad (17)
\]

5. Finally, the back-end server authenticates the attacker as a legal tag and calculates \( R_t^{1+1} \).

### 3.4 Tag traceability attack

I2SRS protocol does not provide enough privacy issue and it is vulnerable to traceability attack which can be applied by an adversary \( \mathcal{A} \), as described below:

**Learning phase:** The attacker sends an execute query \( \langle R, T_0, i \rangle \) and stores the obtained values \( \{M_1^i, \beta_1^i, \gamma_1^i, \alpha_i, R_r^i, V, M_2^i \} \) in this session and blocks the 5.3 step of the protocol to prevent the tag’s updating procedure.

**Challenge phase:** An attacker \( \mathcal{A} \) chooses two fresh tags \( T_0 \) and \( T_1 \) and sends a Test query \( \langle T_0, T_1, i \rangle \). After choosing \( b \in \{0,1\} \) randomly, the attacker takes the tag \( T_b \). Then, he/she starts a new session by generating \( R_t^{i+1} \) randomly and sending an Execute query \( \langle R, T_b, i \rangle \) to the tag \( T_b \), which results in obtaining \( \{M_1^{i+1}, \beta_1^{i+1}, \gamma_1^{i+1}, \alpha_{i+1} \} \).

**Guess phase:** The attacker \( \mathcal{A} \) stops the game \( G \) and announces \( b' \) as his/her guess of \( b \) as:

\[
b' = \begin{cases} 
0 & \text{if } \alpha_i \oplus \beta_i = \alpha_{i+1} \oplus \beta_{i+1} \\
1 & \text{otherwise}
\end{cases}
\]

As a result, we get:

\[
Adv_A^{\text{trace}}(k) = |\text{pr}(b' = b) - \frac{1}{2}| = \left| 1 - \frac{1}{2} \right| = \frac{1}{2} \geq \varepsilon \quad (19)
\]

**Proof:** By paying attention to the structure of protocol that is depicted in Fig. 3, it is obvious that preventing the tag updates its secret values during the \( i \)th session, and presence of \( R_t \), in both \( \alpha \) and \( \beta \) messages, allows an adversary to trace the tag after a successful eavesdropping.

#### 3.5 Tag forward traceability attack

Providing non-forward traceability means, preventing an adversary from tracking the specific tag. Even, the owner of the tag must not be able to trace his/her tag after giving it over. The I2SRS protocol does not provide forward traceability immunity and the attacker can trace the tag \( T_0 \), after \( N \) runs of the protocol (\( \forall N \)). The attack is as follows:

**Learning phase:** In the \( i \)th session of the protocol, an adversary \( \mathcal{A} \) sends a Corrupt query \( \langle T_0, K' \rangle \) and obtains \( \{K_o, P_i, \alpha_i, EPC_s \} \). Weaknesses of updating
process in the I2SRS protocol, lets the attacker calculate \( K_{t+2} \) by performing two times PRNG on \( K_t \).

**Challenge phase:** An attacker \( \mathcal{A} \) chooses two fresh tags \( T_0 \) and \( T_1 \) and sends a Test query \((T_0, T_1, i)\). After choosing \( b \in \{0, 1\} \) randomly, the attacker takes the tag \( T_b \). Now, the attacker starts a new session by sending an Execute query \((R, T_b, i + 2)\) and obtains \( \{M_1^{t+2}, \beta^{t+2}, \gamma^{t+2}, \alpha_1, R_t \} \).

**Guess phase:** The attacker \( \mathcal{A} \) stops the game \( G \) and after calculating \( \{f, g, h\} \), announces \( b' \) as his/her guess of \( b \) as follows:

\[
\begin{align*}
    f &= P(P(K_b^i)) \quad (20) \\
    g &= \beta^{t+2} \oplus f \quad (21) \\
    h &= P(EPC_s \oplus R_t^{i+2} \oplus g) \oplus f \quad (22) \\
    b' &= \begin{cases} 
        0 & \text{if } h = M_1^{t+2} \\
        1 & \text{otherwise}
    \end{cases} \quad (23)
\end{align*}
\]

As a result, we have:

\[
    Adv^{upriv}_{\mathcal{A}}(k) = \left| pr(b' = b) - \frac{1}{2} \right| = \frac{1}{2} \gg \varepsilon \quad (24)
\]

**Proof:** As stated before, the I2SRS protocol suffers from weaknesses in updating technique and transmitting constant messages. The attacker calculates \( K_{t+2} \), like mentioned in learning phase, and obtains \( R_t^{i+2} \) through \( \beta^{t+2} \oplus K_{t+2} \). Moreover, constancy of \( EPC_s \) in all sessions lets the attacker performs as:

\[
    h = P(EPC_s \oplus R_t^{i+2} \oplus g) \oplus f = P(EPC_s \oplus R_t^{i+2} \oplus \beta^{t+2} \oplus f) \oplus f = P\left( EPC_s \oplus R_t^{i+2} \oplus \beta^{t+2} \oplus P\left( P(K_i^b) \right) \right) \oplus P\left( P(K_i^b) \right) = P\left( EPC_s \oplus R_t^{i+2} \oplus \alpha_i \right) \oplus K_{t+2}^i = M_1^{t+2} \quad (25)
\]

3.6 Tag backward traceability attack

Backward traceability immunity assures the owner of the tag that no one will be able to know what happened to him/her before. One of the other vulnerability to the I2SRS protocol in privacy issue is lack of capability to prevent backward traceability attack. Reasons which result in this vulnerability are using the same value for \( EPC_s \) in all sessions and predictability of \( K_{t+1} \) in each session. An adversary \( \mathcal{A} \) performs this attack as follows:

**Learning phase:** An adversary \( \mathcal{A} \) sends a Corrupt query \((T_0, K')\) and obtains \( \{K_t, P_t, \alpha_t, EPC_s\} \). As discussed in Section 3.2, \( K_t \) is a 16 bit string which is one of the elements of \( \{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_{2^{16}}\} \). Therefore, by knowing the updating process of the tag’s secret key, the attacker can guess \( K_{i-1} \) at maximum runs of \( 2^{16} \), which is calculated as:

\[
\begin{align*}
    \text{For } 2 < i < 2^{16} & \quad \text{Choose } K_{i-1} \in \{\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_{2^{16}}\} \\
    & \quad \text{If } K_t \ni P(K_{i-1}) \\
    & \quad \text{Return } K_{i-1} \text{ as a correct secret key}
\end{align*}
\]

(26)

We have the same relations for finding \( P_{i-1} \), which are results of the same inaccuracy in updating procedure. Therefore, the attacker can guess \( P_{i-1} \) at maximum runs of \( 2^{16} \) as calculated:

\[
\begin{align*}
    \text{For } 2 < i < 2^{16} & \quad \text{Choose } P_{i-1} \in \{P_1, P_2, ..., P_{2^{16}}\} \\
    & \quad \text{If } P_t \ni P(P_{i-1}) \\
    & \quad \text{Return } P_{i-1} \text{ as a correct access key}
\end{align*}
\]

(27)

**Challenge phase:** An attacker \( \mathcal{A} \) chooses two fresh tags \( T_0 \) and \( T_1 \) and sends a Test query \((T_0, T_1, i)\). After choosing \( b \in \{0, 1\} \) randomly, the attacker takes the tag \( T_b \), sends an Execute query \((R, T_b, i - 1)\) and obtains \( \{M_1^{i-1}, \beta^{i-1}, \gamma^{i-1}, \alpha_{i-1}, R_{t-1}^{i-1}\} \).

**Guess phase:** The attacker finishes the game \( G \) and proclaims \( b' \) as his/her guess of \( b \) as follows:

\[
    b' = \begin{cases} 
        0 & \text{if } M_2^{i+2} = P(EPC_s \oplus R_t^{i}) \oplus R_t \\
        1 & \text{otherwise}
    \end{cases}
\]

(28)

As a result, we have,

\[
    Adv^{upriv}_{\mathcal{A}}(k) = \left| pr(b' = b) - \frac{1}{2} \right| = \frac{1}{2} \gg \varepsilon \quad (29)
\]

**Proof:** Choosing a correct value for \( K_{i-1} \), let the attacker calculates \( R_t^i \) in the \((i-1)\)th session by computing \( \beta^{i-1} \oplus K_{i-1} \). Then, \( M_2^{i+2} \) will be computable, if he/she knows the precise value for the access key. Subliminally, the attacker is able to perform backward traceability attack at maximum runs \( 2 \times 2^{16} = 2^{17} \) computations.

4. Improvements of Cai et al.’s protocols

4.1 Improvements of the IHRMA protocol

Structure of generating \( RID_i \) can be highlighted as the greatest weak point in IHRMA protocol. So, we improve the IHRMA protocol as depicted in Fig 4. By changing the messages \( RID_i \) to \( H(S_j \oplus R_t) \), the attacker is prevented to access this message. One of the other vulnerability in the IHRMA protocol is the manner of producing and transmitting \( C_3, C_4 \) and \( C_5 \).
| Back-end Server \((ID_k, S_j, S_{j-1}, DATA)\) | Reader | Tag \((ID_k, S_j)\) |
|---|---|---|
| 1.1 Request: \(R_r\) | | 2.1 \(R_{ID_1} = H(S_j \oplus R_j)\) |
| | 1.2 | 2.2 \(C_1 = H(ID_k \oplus R_1 \oplus R_{ID_1})\) |
| 4.1 For each tuple \((ID_k; S_j)\) | | 2.3 \(C_2 = H(S_j(0:47) \parallel ID_k(48:95))\) |
| | Calculate \(C_2 = S_j(0:47) \parallel ID_k(48:95)\) | 2.4 \(C_1, C_2 \oplus R_t\) |
| | Calculate \(R_t^i\) | 3.1 \(C_3 = C_1, C_2 \oplus R_t\) |
| | Compute \(\text{RID}_i^j\) | |
| | verify \(C_i^j = C_i\) | |
| | if it does not match | |
| | \(\text{repeat with tuple } (ID_k; S_{j-1})\) | |
| 4.2 calculate \(R_t^i = C_2 \oplus C_3\) | | |
| 4.3 calculate \(\text{RID}_i^j\) | | |
| 4.4 calculate \(C_i^j = H(ID_k \oplus R_1 \oplus R_2 \oplus R_{ID_1})\) | | |
| 4.5 calculate \(C_3 = H(\text{RID}_1 \parallel C_2)\) | | |
| 4.6 updating | | |
| \(S_j - 1 \leftarrow S_j\) | | |
| \(S_j \leftarrow S_j + 1\) | | |

Fig. 4. Improved IHRMA protocol

messages, which let the attacker leave the protocol unfinished via varying \(R_t\). By omitting \(C_4\) and \(C_5\) messages through our improved protocol and generating \(C_3\) as \(H(C_2 \parallel \text{RID}_1)\), an adversary is not able to perform DoS attack.

4.1.1 Tag/Reader impersonation resistance

Presence of the \textit{mod} operator in the IHRMA protocol causes the occurrence of \textit{Notion 1}. In our improved protocol, we generate \(\text{RID}_1\) via a hash function which prevents the attacker from omitting \(R_t\) in \(\text{RID}_1\). Moreover, the attacker will not be able to substitute \(R_t^{i+1}\) with \((R_t^i \oplus R_t \oplus R_{t+1}^i)\), cause that our new messages are preserved by hash functions.

4.1.2 DoS attack resistance

Presence of \(R_t\) in \(C_4\) and \(C_5\) messages through the IHRMA protocol, let the attacker XOR a random number with them, which prevent the tag authenticate the back-end server and the reader as a legal parts of RFID system. In our improved protocol we generate \(C_3\) as \(H(\text{RID}_1 \parallel C_2)\) and by omitting \(C_4\) and \(C_5\) messages, \(C_3\) is the only oracle sends to the tag for authentication. Therefore, an adversary will not be able to alter messages as he/she desires, which provides a secure and assured authentication procedure.

4.2 Improved version of I2SRS protocol

Cai et al. [18] believe that Yoon’s protocol [13] does not provide the privacy of end-user, so they improved it and proposed the I2SRS protocol. As we described in Section 3, the I2SRS protocol is still vulnerable to different attacks, so this Section, propose a strengthened versions of Cai et al.’s protocol to overcome its weaknesses, which is shown in Fig. 5. Also, the security and privacy analysis of our proposed protocol is provided.

It can be considered that using the \(R_t \oplus R_r\) in producing \(M_1\), is one of the main weaknesses in the I2SRS protocol, which helps the attacker to calculate \(\gamma\) and \(\beta\). Moreover, updating procedure is the other weaknesses which result in traceability attacks. Therefore, we improve the I2SRS protocol by substituting \(M_1\), \(\alpha_i\) and \(\gamma_i\) messages as follows,

\[
M_1 = P(EPC_s \oplus R_t) \oplus P(R_t) \oplus K_i \quad (30)
\]

\[
\alpha_i = \alpha_i \oplus R_{\text{new}} \quad (31)
\]

\[
\gamma = R_t \oplus P(\alpha_i \oplus K_i) \oplus P_i \quad (32)
\]

\(R_{\text{new}}\) is a new random number generated by the tag. Moreover, the updating procedure will alter with,

\[
K_{i+1} \leftarrow P(K_i \oplus R_{\text{new}}) \quad (33)
\]
After the presented analysis, a comparison of the I2SRS protocols with some similar protocols are presented in Table 1 and 2, respectively. As it was mentioned in Section 3, I2SRS protocol provides privacy by defining a new format for transferred massages through impersonating a legal tag or reader, in replay attacks. In our improved I2SRS protocol, usage of a new random number, $R_{\text{New}}$, and change the structure of produced messages, makes it impossible for an adversary to re-use the eavesdropped ones through last sessions.

### 4.2.1 Secret reveal resistance

We showed that weaknesses in creation of $\alpha_i$ and its updating manner, will result in achieving the correct $K_i$ by the attacker. But in our improved I2SRS protocol, changing $\alpha_i$ to $P(R_t \oplus R_t \oplus P_i)$, prevents the attacker from obtaining $R_t$ and $K_i$.

### 4.2.2 Replay attack resistance

An adversary $\mathcal{A}$ tries to verify, denies or omits the transferred massages through impersonating a legal tag or reader, in replay attacks. In our improved protocol, usage of a new random number, $R_{\text{New}}$, and change the structure of produced messages, makes it impossible for an adversary to re-use the eavesdropped ones through last sessions.

### 4.2.3 Impersonation attack resistance

An adversary $\mathcal{A}$ is necessitous of knowing the values of $K_i, R_t$ and $EPC_s$ to compute $M_1$, $\beta$ and $\gamma$ messages. By defining a new format for $M_1$ and $\alpha_i$, an adversary is not able to access the secret value of the improved I2SRS protocol. Besides, usage of the last stored $M_1$ is not profitable for him/her in the current session.

### 5. Results and comparisons

#### 5.1 Privacy guarantee

The improved I2SRS protocol provides privacy through changing the method of updating procedure. Moreover, it prevents backward traceability, traceability and forward traceability attacks via defining the updated $K_i$ as $P(K_i \oplus R_{\text{new}})$.

After the presented analysis, a comparison of the privacy and the security of the improved HRMA and I2SRS protocols with some similar protocols are presented in Table 1 and 2, respectively. As it
can be seen, the proposed protocols are protected against numerous attacks such as traceability, backward traceability, forward traceability, impersonation and DoS. Our proposed protocols provide secure and untraceable communications for RFID elements.

6. Conclusions

RFID applications are developing in different areas, which provide comfortability, rapidity and accuracy, but an important issue that must be considered is the assurance of a secure and private connection during the communication procedure. This paper investigated the performance and vulnerabilities of two recent authentication protocols proposed based on R-RAPSE rules. Based on Ouafi-Phan formal privacy model, it is shown that both the IHRMA and the I2SRS protocols cannot provide private authentication for RFID users. To enhance the proficiency of these two protocols, we proposed some improvements in this research. This paper proved that the proposed protocols provide required privacy and security against various types of attacks and can solve the drawbacks of the discussed works.

References

[1] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things," in Proceedings of the 14th ACM Workshop on Hot Topics in Networks, p. 5, 2015.
[2] L. Batina, J. Guajardo, B. Preneel, P. Tuyls, and I. Verbauwhede, "Public-key cryptography for RFID tags and applications," in RFID Security, ed: Springer, pp. 317-348, 2008.
[3] S. M. Alavi, B. Abdolmaleki, and K. Baghery, "Vulnerabilities and Improvements on HRAP+, a Hash-Based RFID Authentication Protocol," Advances in Computer Science: an International Journal, vol. 3, pp. 51-56, 2014.
[4] K. Baghery, B. Abdolmaleki, and M. J. Emadi, "Game-based cryptanalysis of a lightweight CRC-based authentication protocol for EPC tags," mirkabir International Journal of Electrical & Electronics Engineering (AIJ-EEE), vol. 46, pp. 27-36, 2014.
[5] E. Pagnin, C. Dimitrakakis, A. Abidin, and A. Mitrokotsa, "On the Leakage of Information in Biometric Authentication," in Progress in Cryptology--INDOCRYPT 2014, ed: Springer, pp. 265-280, 2014.
[6] T.-C. Yeh, Y.-J. Wang, T.-C. Kuo, and S.-S. Wang, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard," Expert Systems with Applications, vol. 37, pp. 7678-7683, 2010.
[7] S. S. S. Ghaemmaghami, M. Mirmohseni, and A. Haghibin, "A Privacy Preserving Improvement for SRTA in Telecare Systems," arXiv preprint arXiv:1510.04197, 2015.
[8] G. Avoine, "Cryptography in Radio Frequency Identification and Fair Exchange Protocols," PhD. Thesis, Swiss Federal Institute of Technology, Switzerland, 2005.
[9] E. Pagnin, A. Yang, G. Hancke, and A. Mitrokotsa, "HB+ DB, mitigating man-in-the-middle attacks against HB+ with distance bounding," in Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, p. 3, 2015.
[10] H.-Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity,"
[11] K. Fan, N. Ge, Y. Gong, H. Li, R. Su, and Y. Yang, "ULRAS: Ultra-Lightweight RFID Authentication Scheme for Mobile Device," in Wireless Algorithms, Systems, and Applications, ed: Springer, pp. 114-122, 2015.

[12] M. Safkhani, N. Bagheri, P. Peris-Lopez, and A. Mitrokotsa, "On the traceability of tags in SUAP RFID authentication protocols," in RFID-Technologies and Applications (RFID-TA), 2012 IEEE International Conference on, pp. 292-296, 2012.

[13] E.-J. Yoon, "Improvement of the securing RFID systems conforming to EPC class 1 generation 2 standard," Expert Systems with Applications, vol. 39, pp. 1589-1594, 2012.

[14] M. Mohammadi, M. Hosseinzadeh, and M. Esmaeildoust, "Analysis and improvement of the lightweight mutual authentication protocol under EPC C-1 G-2 standard," Advances in Computer Science: an International Journal, vol. 3, pp. 10-16, 2014.

[15] S. Abdolmaleky, S. Atapoor, M. Hajighasemlou, and H. Sharini, "A Strengthened Version of a Hash-based RFID Server-less Security Scheme," Advances in Computer Science: an International Journal, vol. 4, pp. 18-23, 2015.

[16] B. Abdolmaleki, H. Bakhsi, K. Baghery, and M. R. Aref, "Analysis of an RFID authentication protocol in accordance with EPC standards," International Journal of Information & Communication Technology Research, vol. 6, pp. 7-12, 2014.

[17] J.-S. Cho, S.-S. Yeo, and S. K. Kim, "Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value," Computer Communications, vol. 34, pp. 391-397, 2011.

[18] Q. Cai, Y. Zhan, and J. Yang, "The Improvement of RFID Authentication Protocols Based on R-RAPSE," Journal of Networks, vol. 9, pp. 28-35, 2014.

[19] G. Avoine, "Adversarial Model for Radio Frequency Identification," IACR Cryptology ePrint Archive, vol. 2005, p. 49, 2005.

[20] A. Juels and S. A. Weis, "Defining strong privacy for RFID," ACM Transactions on Information and System Security (TISSEC), vol. 13, p. 7, 2009.

[21] C. H. Lim and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer," in Information and Communications Security, ed: Springer, pp. 1-20, 2006.

[22] S. Vaudenay, "On privacy models for RFID," in Advances in Cryptology--ASIACRYPT 2007, ed: Springer, pp. 68-87, 2007.

[23] K. Ouafi and R. C.-W. Phan, "Privacy of recent RFID authentication protocols," in Information Security Practice and Experience, ed: Springer, pp. 263-277, 2008.

[24] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for RFID privacy," in Computer Security--ESORICS 2010, ed: Springer, pp. 1-18, 2010.

[25] D. Moriyama, S. i. Matsuo, and M. Ohkubo, "Relations among notions of privacy for RFID authentication protocols," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 97, pp. 225-235, 2014.

[26] G. Avoine, I. Coisel, and T. Martin, "Untraceability model for RFID," Mobile Computing, IEEE Transactions on, vol. 13, pp. 2397-2405, 2014.