The ISC Int'l Journal of Information Security
ISeCure

July 2015, Volume 7, Number 2 (pp. 135–149)
http://www.isecure-journal.org

Enhancing Privacy of Recent Authentication Schemes for Low-Cost RFID Systems☆

Karim Baghery [1,*], Behzad Abdolmaleki [1], Bahareh Akhbari [2,*], and Mohammad Reza Aref [3]

[1] *Information Systems and Security Lab (ISSL), Sharif University of Technology, Tehran, Iran.*
[2] *Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran.*
[3] *ISSL Lab, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 14 January 2015
Revised: 1 August 2015
Accepted: 10 October 2015
Published Online: 16 October 2015

Keywords:
RFID Authentication Protocol, Security, Privacy, EPC C1 G2 Standard.

ABSTRACT

Nowadays Radio Frequency Identification (RFID) systems have appeared in lots of identification and authentication applications. In some sensitive applications, providing secure and confidential communication is very important for end-users. To this aim, different RFID authentication protocols have been proposed, which have tried to provide security and privacy of RFID users. In this paper, we analyze the privacy of two recently proposed RFID authentication protocols in 2012 and 2013. We present several traceability attacks including traceability, backward traceability and forward traceability against the first protocol. We also show that, the second protocol not only suffers from Denial-of-Service (DoS) attack, but also it is vulnerable to traceability and backward traceability attacks. We present our privacy analysis based on a well-known formal RFID privacy model which has been proposed by *Ouafi* and *Phan* in 2008. Then, in order to overcome the weaknesses, we apply some modifications on these protocols and propose two modified versions.

1 Introduction

Radio Frequency Identification (RFID) technology is widely recognized as a prominent method to provide fast and precise authentication and identification for different applications in proximity and vicinity areas [1]. In addition, RFID systems are interesting candidates to be implemented in the next generation of internet, which is called Internet of Things (IoT)[2]. The IoT systems allow objects and people to make a connection at anyplace and anytime via any sensing devices, which can exchange data between two objects [3]. Therefore, the mobile RFID readers can play the role of IoT gateway.

Generally, RFID systems consist of a large number of tags, readers and a back-end server [4]. A typical model of an RFID system is depicted in Figure 1 RFID systems use RF technology to provide wireless communication between the tags and the readers for different identification and authentication applications. The tag is an electronic chip equipped with microstrip antenna to setup a wireless connection with the reader. In different applications, different types of information are stored in the RFID tags. In some cases, the tag just contains a unique identification code like an Electronic Product Code (EPC). In this case, the

---

☆ This article is an extended version of an ISCISC'14 paper.
* Corresponding author.

Email addresses: `k.baghery.1988@ieee.org` (K. Baghery), `b.abdolmaleki.ir@ieee.org` (B. Abdolmaleki), `akhbari@kntu.ac.ir` (B. Akhbari), `aref@sharif.edu` (M. R. Aref)

ISSN: 2008-2045 © 2015 ISC. All rights reserved.

identification code is written onto the tag and it is not modifiable (i.e. it is read only). In some applications the tag has a memory that can be modified or erased by a legal user (readable/writeable) [4]. Based on power supply sources, the RFID tags are classified into three different classes including active, passive and semi-passive tags [4]. The next part of an RFID system is the reader that is located between the tag and the back-end server and acts as an interrogator (shown in Figure 1). In other words, it exchanges some messages between the tag and the back-end server and makes data accessible to the tag. The main part of an RFID system is the database or the back-end server. All secret values and some necessary data of the tags are stored in the back-end server and it uses them for identification and authentication processes [5].

Although RFID systems provide user-friendly services and are one of the most popular technologies in different authentication applications, they may suffer from some security and privacy concerns. These systems may be susceptible to different security and privacy attacks such as *Denial-of-Service* (DoS), *Man-in-Middle* (MiM), *Impersonation*, *Reveal Secret Parameter* and different *Traceability* attacks [5]. As RFID systems have been deployed in different parts of our daily life, without proper protection RFID systems can make privacy concerns for end-users [6]. In the following, we review the concepts of untraceability, backward untraceability, and forward untraceability which are three essential issues in providing privacy for RFID users.

- **Untraceability:** always the end-user's privacy is a prominent issue in the applications of novel technologies. Likewise, in the RFID systems, it is very important that the attacker should be unable to trace a specific tag, in case that he/she has access to the exchanged messages between the tag and a valid reader before last successful authentication. Namely, an RFID tag is untraceable if its responses to two consecutive runs, are uncorrelated [7].
- **Forward Untraceability:** an RFID authentication protocol which provides forward untraceability is able to prevent tracing the location of a specific tag in the future runs. More precisely, if an attacker corrupts secret keys of a specific tag, it is impossible for the attacker to track the location of the tag in the future sessions [8].
- **Backward Untraceability:** another goal of an RFID authentication protocol is to provide backward untraceability [6]. To this aim, in an RFID system if an attacker obtains the current exchanged messages between the tag and the reader, he/she should be unable to trace the location of a specific tag in the previous session. This goal can be achieved by proper updating of the tag's secret keys.

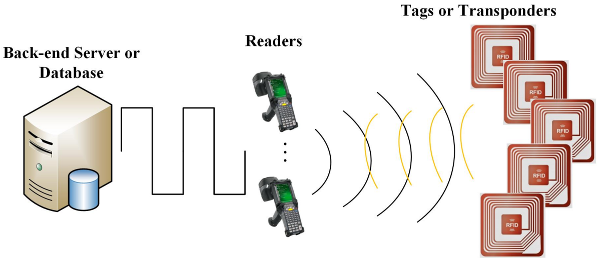


Figure 1. A System model of RFID systems

It is undeniable that a secure and confidential RFID authentication protocol can prevent many security and privacy concerns [9]. In the last few years, there has been a large amount of literature on RFID authentication protocols [4], [9–17]. On the other hand, Electronic Product Code Class 1 Generation 2 (EPC C1 G2) standard [18] is one of the popular standards which recently has got more attention. Actually lots of RFID authentication protocols have been proposed that are compliant with EPC standards [19–24]. It also should be noted that, due to some restrictions on memory and computation limitations of RFID tags, RFID authentication schemes are designed by lightweight cryptographic operators [5].

In 2007, *Chien* and *Chen* [19] proposed an improved RFID authentication protocol which is a refined version of Duc *et al.*'s protocol [25] and *Karthikeyan-Nesterenko*'s protocol [26]. In the improved protocol, *Chien* and *Chen* proposed two main modifications in the structure of the analyzed protocols. The first modification is updating the secret keys of the back-end server and the tag after each successful authentication, and the second one is storing both the old and new secret keys in the back-end server, which causes the improved protocol to be more efficient against DoS attack. Also, updating the secret keys increases the forward secrecy significantly. *Chien* and *Chen*'s protocol [19] is proposed for EPC compliant tags and in order to protect exchanged messages between the tag, the reader and the back-end server, the Exclusive OR (XOR), Pseudo Random Number Generator (PRNG) and Cyclic Redundancy Code (CRC) operations have been utilized. However, in 2010, Yeh *et al.* [22] showed that *Chien* and *Chen*'s protocol is not safe against DoS attack and also it has a privacy weakness which stemmed from improper usage of CRC operator. Then, in order to omit the mentioned problems, Yeh *et al.* applied some modifications on *Chien* and *Chen*'s protocol and proposed an improved RFID authentication protocol which is under EPC C1 G2 standard as well. Although, Yeh *et al.* have tried to provide secure communications for RFID end-users, in 2012 *Yoon* discovered two flaws in the structure of Yeh *et al.*'s protocol. *Yoon* illustrated that Yeh *et al.*'s protocol has data integrity problem, and also it cannot provide forward secrecy [20] . Then, he proposed a modified version of Yeh *et al.*'s protocol and claimed that it eliminates the mentioned weaknesses.

Generally the privacy of RFID authentication protocols can be analyzed based on *ad-hoc* methods and *formal* methods [5]. In the ad-hoc approaches, an adversary defines some notations and analyzes the privacy of a protocol based on the defined notations. In other words, the adversary performs his/her operations and computations based on informal methods which are not valid as much as formal methods [27]. On the other hand, in the formal approaches, the attacker has various controls over communication channels which are defined in specific queries. More precisely, an attacker has various abilities which are classified into different categories and can be used in both active and passive attacks [7]. In order to discover all drawbacks of RFID authentication protocols it is essential to use a formal RFID privacy model [28]. In the last decade, different RFID formal privacy models have been proposed [6], [27], [29–33]. In this paper, we present our privacy analysis against *Yoon* and Jung *et al.*'s protocols based on a well-known *Ouafi* and *Phan* formal privacy model which is presented in [31]. *Ouafi* and *Phan*'s privacy model is a well-known game-based RFID privacy model and is one of the highly cited models which have been proposed in the recent years.

In 2011, Safkhani *et al.* [34] cryptanalyzed *Yoon*'s protocol and showed that *Yoon*'s protocol has some security and privacy weaknesses. They analyzed the privacy of *Yoon*'s protocol based on *ad-hoc* methods and presented a traceability attack against *Yoon*'s protocol. In addition, in [35] *Mohammadali et al.* showed that *Yoon's* protocol has several security problems and also they presented an ad-hoc traceability attack against the *Yoon*'s protocol which is different from the presented attack in [34]. Both of these attacks result from a weakness in the tag responses of *Yoon*'s protocol. Continuing on our seminal work [36], this paper formally analyses the privacy of *Yoon*'s protocol. We analyze the privacy of *Yoon*'s protocol based on a formal RFID privacy model and show that the privacy of this protocol is not provided, and an attacker can trace the location of a specific tag. More precisely, we formally show that *Yoon*'s protocol is not resistant against various traceability attacks including traceability, backward traceability and forward traceability.

Another approach for providing security and privacy of RFID users is using hash functions in authentication protocols [37–42]. In 2013, Jung *et al.* investigated three hash-based RFID authentication protocols which have been proposed in [37–39] and proposed a novel Keyed-hash based Message Authentication Code (HMAC) RFID mutual authentication protocol [40]. Jung *et al.* analyzed their proposed protocol against various security and privacy attacks including *DoS*, *Impersonation* and *Traceability* attacks, and claimed that their protocol resists against all these attacks and can provide users' security and privacy [40]. However, in this study, we show that Jung *et al.*'s protocol still has some security and privacy flaws and suffers from DoS attack, traceability attack and backward traceability attack.

Moreover, in order to overcome all the mentioned weaknesses and increasing the performance of analyzed protocols, we apply some modifications on the analyzed protocols and propose strengthened versions of Yoon and Jung *et al.*'s protocols. Our analyses show that the improved protocols are resistant against various attacks and they can provide security and confidentiality for RFID users. Moreover, the security and the privacy of the improved protocols are compared with some similar authentication protocols which are proposed for RFID systems.

The reminder of this paper is organized as follows: Section 2 introduces *Ouafi* and *Phan*'s formal privacy model which is used in our privacy analysis. *Yoon*'s protocol and its privacy analysis are provided in Section 3. In Section 4, Jung *et al.*'s protocol and its weaknesses are given. Our enhancements on *Yoon*'s protocol and Jung *et al.*'s protocol are reported in Section 5. Also in this section, the proposed protocols are compared with respect to security and privacy with some existing protocols. Finally, we conclude the paper in Section 6.

2 Ouafi and Phan privacy model

In 2008, *Ouafi* and *Phan* [31] presented a formal privacy model which is used to evaluate RFID authentication protocols. The *Ouafi* and *Phan* privacy model is summarized as follows.

In this model, the attacker $\mathcal{A}$ can eavesdrop on all channels between tags and readers and also it can perform active and passive attacks against them. As well, the attacker $\mathcal{A}$ is allowed to run the following queries:

(1) ***Execute query* $(\mathbf{R}, \mathbf{T}, \mathbf{i})$:** Passive attacks take place in this query. In other words, the attacker can eavesdrop on all transmitted messages between the tag T and the reader R in the ith session. As a result, the attacker obtains all exchanged data between the tag T and the reader R.

(2) ***Send query* $(\mathbf{U}, \mathbf{V}, \mathbf{m}, \mathbf{i})$:** This query models

an active attack in RFID systems. In this query, the attacker $\mathcal{A}$ has permission to impersonate the reader U in the ith session, and forwards the message m to the tag V. In addition, 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.

(3) ***Corrupt query* (T, K$'$):** In this query, the attacker $\mathcal{A}$ has permission to access secret keys of the tag. In fact, the attacker $\mathcal{A}$ has physical access to the tag's database. In addition, the attacker $\mathcal{A}$ can set the secret key to K'.

(4) ***Test query* ($\mathbf{T_0}$, $\mathbf{T_1}$, i):** When this query is executed in the particular session i, after completing the ith session, a random number bit $b \in \{0, 1\}$ is generated by the challenger and it is delivered $T_b \in \{T_0, T_1\}$ to the attacker. Now, the attacker succeeds if he/she can guess the bit b, correctly.

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

(1) ***Learning phase:*** The attacker $\mathcal{A}$ has permission to send each one of the queries such as *Execute*, *Send* and *Corrupt*, and interact with the reader R and T_0, T_1 that are chosen randomly.

(2) ***Challenge phase:*** The attacker $\mathcal{A}$ selects two tags T_0 and T_1 and forwards a *Test query* (T_0, T_1, i) to the challenger. After that, the challenger selects $b \in \{0, 1\}$ randomly and the attacker $\mathcal{A}$ determines a tag $\mathrm{T}_b \in \{\mathrm{T}_0, \mathrm{T}_1\}$ using *Execute* and *Send* queries.

(3) ***Guess phase:*** Eventually, the attacker $\mathcal{A}$ finishes the game G and outputs a bit $b' \in \{0, 1\}$ as a guess of b.

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

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

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

In the rest of paper, using privacy model of *Ouafi* and *Phan*, privacy of *Yoon*'s and Jung *et al.*'s protocols are investigated.

3 Privacy Analysis of Yoon's Protocol

This section aims to analyze the privacy of *Yoon*'s protocol against various traceability attacks. It is shown that Yoon's protocol has some weaknesses which make it vulnerable to all traceability attacks including traceability, backward traceability and forward traceability attacks. Before presenting the privacy analysis, firstly we introduce *Yoon*'s protocol that proposed in [20].

3.1 Yoon's Protocol

In [20], *Yoon* proposed an improved mutual authentication protocol for RFID systems which conforms to EPC C1 G2 standard. The notations that are used in *Yoon*'s protocol are shown in Table 1. The structure of *Yoon*'s protocol that is shown in Figure 2 can be summarized as follows,

Table 1. The Notations of Yoon's Protocol

Notation	Description
EPC_s	A 16-bit Electronic Product Code
DATA	The corresponding record for the tag kept in the back-end server
K_i	The authentication key stored in the tag to be used by database to authenticate the tag at the $(i+1)^{th}$ authentication phase
P_i	The access key stored in the tag to be used by database to authenticate the tag at the $(i+1)^{th}$ authentication phase
C_i	The database index stored in the tag to find the corresponding record of the tag in the database
P_{old}	The old access key stored in the database
P_{new}	The new access key stored in the database
K_{old}	The old authentication key stored in the database
K_{new}	The new authentication key stored in the database
C_{old}	The old database index stored in the database
C_{new}	The new database index stored in the database
N_d	The 16-bit random number that generated by device d
PRNG	Pseudo random number generator
H(·)	Hash function
RID	The reader identification number
A⊕B	Message A is XORed with message B

a) Initial phase

In this phase, some initial secret values such as K_0, P_0 and C_0 that are generated randomly in the manufacture, are shared between the tag and the back-end server. Also, the corresponding values of the mentioned parameters in the back-end server are set to these initial values ($K_{old} = K_{new} = K_0$, $P_{old} = P_{new} = P_0$ and $C_{old} = C_{new} = C_0$).

b) Authentication phase

This phase includes five steps as follows,

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

Step 2. Tag → Reader: Upon receiving N_R, the tag generates a random number N_T. It computes the following messages and sends them along C_i to the reader.

$$\begin{aligned} M_1 &= PRNG\left(EPC_s \oplus N_R \oplus N_T\right) \oplus K_i, \\ D &= N_T \oplus K_i \\ E &= N_T \oplus PRNG(C_i \oplus K_i). \end{aligned}$$

Step 3. Reader → Back-end server: The reader calculates $V = H(RID \oplus N_R)$ and forwards the messages $(M_1, D, C_i, E,\ N_R, V)$ to the back-end server.

Step 4. Back-end server → Reader: Based on the received messages from the reader, the back-end server performs the following operations,

(1) The back-end server verifies $V \overset{?}{=} H(RID \oplus N_R)$ and follows the rest of authentication procedure.

(2) The back-end server first computes $I_X = M_1 \oplus K_X$ for $X \in \{old,\ new\}$. Then it checks whether $I_X = PRNG(EPC_s \oplus N_R \oplus D \oplus K_X)$ and determines that $X = old$ or new.

(3) Now by using the obtained $X = old$ or new, the back-end server verifies $E \overset{?}{=} N_T \oplus PRNG(C_X \oplus K_X)$. If $E = N_T \oplus PRNG(C_X \oplus K_X)$, it authenticates the tag and responds to the reader by the following messages,

$$\begin{aligned} M_2 &= PRNG(EPC_s \oplus N_T) \oplus P_X \\ Info &= DATA \oplus RID \\ MAC &= H(DATA \oplus N_R), \end{aligned}$$

otherwise, the back-end server aborts the protocol.

(4) Finally, the back-end server updates its secret values as follows,

```
$If \quad X = new$
    $K_{old} \leftarrow K_{new} \leftarrow PRNG\left(K_{new}\right)$
    $P_{old} \leftarrow P_{new} \leftarrow PRNG\left(P_{new}\right)$
    $C_{old} \leftarrow C_{new} \leftarrow PRNG\left(N_T \oplus N_R\right)$
$Else$
    $C_{new} \leftarrow PRNG\left(N_T \oplus N_R\right)$
$End$
```

Step 5. Reader → Tag: Now using the received message $Info$, the reader computes $DATA = Info \oplus RID$, verifies $H\left(DATA \oplus N_R\right) \overset{?}{=} MAC$, and then sends M_2 to the tag.

Finally, utilizing the received message M_2, the tag verifies $M_2 \oplus P_i \overset{?}{=} PRNG(EPC_s \oplus N_T)$. If the answer is Yes, the tag updates its secret values by,

$$\begin{aligned} K_{i+1} &\leftarrow PRNG\left(K_i\right) \\ P_{i+1} &\leftarrow PRNG\left(P_i\right) \\ C_{i+1} &\leftarrow PRNG\left(N_T \oplus N_R\right), \end{aligned}$$

otherwise, the tag aborts the protocol.

3.2 Traceability Attack

Providing an untraceable communication for end-users is one of the primary goals for each RFID authentication protocol. In this subsection we aim to show that *Yoon*'s protocol cannot protect RFID users against traceability attack. To reach this aim, we show that an attacker can act as follows,

Learning phase: In round (i), the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$ by sending N_R, and he/she obtains $C_i^{T_0}$.

Challenge phase: The attacker $\mathcal{A}$ selects two new tags T_0 and T_1, and sends a $Test\ query\ (T_0,\ T_1,\ i+1)$. 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, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+1)$ by sending N_R, and he/she obtains $C_{i+1}^{T_b}$.

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

$$b' = \begin{cases} 0 & if\ C_{i+1}^{T_b} = C_i^{T_0} \\ 1 & otherwise \end{cases}$$

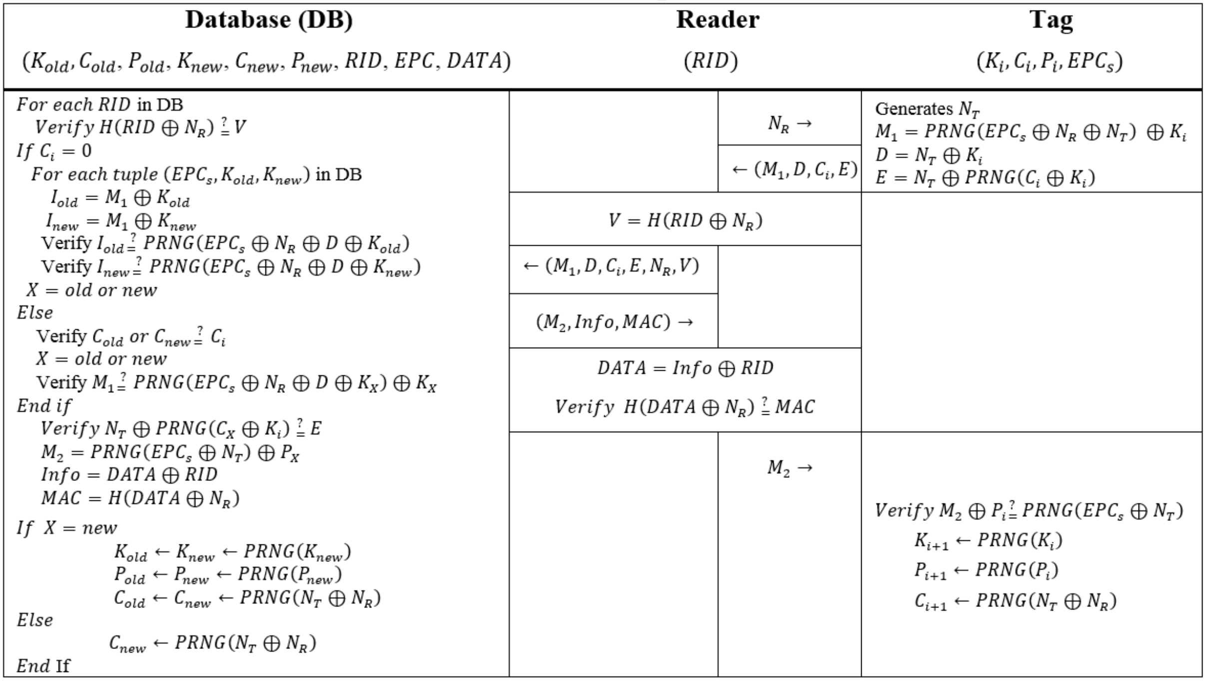
As a result,

$$\begin{aligned} Adv_{\mathcal{A}}^{upriv}(K) &= \left|pr\left(b' = b\right) - pr\left(random\ coin\ flip\right)\right| \\ &= \left|pr\left(b' = b\right) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \epsilon. \end{aligned}$$

Proof: In *Yoon*'s protocol, according to Figure 2, the following equation can be written.

$$\begin{aligned} If\ \ T_b = T_0 \Longrightarrow \quad C_{i+1}^{T_b} &= PRNG\left(N_{T,i}^{T_0} \oplus N_{R,\ i}^{T_0}\right) \\ &= C_i^{T_0} \end{aligned}$$

Note that, the tag T_0 does not update its secret values in the *Learning phase* and uses the same secret value C_i in both *Learning* and *Challenge* phases.



Database (DB)	Reader		Tag
$(K_{old}, C_{old}, P_{old}, K_{new}, C_{new}, P_{new}, RID, EPC, DATA)$	(RID)		(K_i, C_i, P_i, EPC_s)
For each RID in DB $Verify\ H(RID \oplus N_R) \overset{?}{=} V$ $If\ C_i = 0$ $For\ each\ tuple\ (EPC_s, K_{old}, K_{new})$ in DB $I_{old} = M_1 \oplus K_{old}$ $I_{new} = M_1 \oplus K_{new}$ Verify $I_{old} \overset{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{old})$ Verify $I_{new} \overset{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{new})$ $X = old\ or\ new$ $Else$ Verify $C_{old}\ or\ C_{new} \overset{?}{=} C_i$ $X = old\ or\ new$ Verify $M_1 \overset{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_X) \oplus K_X$ $End\ if$ $Verify\ N_T \oplus PRNG(C_X \oplus K_i) \overset{?}{=} E$ $M_2 = PRNG(EPC_s \oplus N_T) \oplus P_X$ $Info = DATA \oplus RID$ $MAC = H(DATA \oplus N_R)$ $If\ \ X = new$ $K_{old} \leftarrow K_{new} \leftarrow PRNG(K_{new})$ $P_{old} \leftarrow P_{new} \leftarrow PRNG(P_{new})$ $C_{old} \leftarrow C_{new} \leftarrow PRNG(N_T \oplus N_R)$ $Else$ $C_{new} \leftarrow PRNG(N_T \oplus N_R)$ End If		$N_R \rightarrow$ $\leftarrow (M_1, D, C_i, E)$	Generates N_T $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$ $D = N_T \oplus K_i$ $E = N_T \oplus PRNG(C_i \oplus K_i)$
	$V = H(RID \oplus N_R)$		
	$\leftarrow (M_1, D, C_i, E, N_R, V)$		
	$(M_2, Info, MAC) \rightarrow$		
	$DATA = Info \oplus RID$ $Verify\ \ H(DATA \oplus N_R) \overset{?}{=} MAC$		
		$M_2 \rightarrow$	$Verify\ M_2 \oplus P_i \overset{?}{=} PRNG(EPC_s \oplus N_T)$ $K_{i+1} \leftarrow PRNG(K_i)$ $P_{i+1} \leftarrow PRNG(P_i)$ $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$

Figure 2. The Yoon's Protocol [20].

3.3 Backward Traceability Attack

This section shows that there is another privacy concern in *Yoon*'s protocol which is vulnerability against backward traceability attack. This weakness is caused due to a flaw in the updating of secret key K_i which is PRNG of K_{i-1}. By considering this fact, an attacker can obtain K_{i-1} with maximum 2^{16} computations which is given with more details as follows.

Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $K_i^{T_0}$ from the tag T_0. Now, since K_i is a 16-bit string, thus $K_i \in U$ where $U = \{u_1, u_2, \dots, u_{2^{16}}\}$. Now,

```
$For\ 1 \leq j \leq 2^{16}$
    $Choose\ u_j \in U$
    $if\ \ K_i^{T_0} = PRNG\,(u_j)\ \ then$
    $return\ u_j\ as\ K_{i-1}^{T_0}$
$End$
```

It can be seen that the value of $K_{i-1}^{T_0}$ can be obtained.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for test, and sends a $Test\ query\ (\ T_0,\ T_1, i)$. 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 $(i-1)$th, the attacker $\mathcal{A}$ sends an $Execute\ query\,(R,\ T_b, i-1)$, and obtains $C_{i-1}^{T_b}$, $D_{i-1}^{T_b}$ and $E_{i-1}^{T_b}$.

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 determine $b' \in \{0,\ 1\}$, the attacker uses the following rule.

$$b' = \begin{cases} 0\ \ if\ \ E_{i-1}^{T_b} \oplus D_{i-1}^{T_b} = K_{i-1}^{T_0} \oplus PRNG\left(K_{i-1}^{T_0} \oplus C_{i-1}^{T_b}\right) \\ 1\ \ otherwise \end{cases}$$

So, $Adv_{\mathcal{A}}^{upriv}\,(k)$ is computed as follows:

$$Adv_{\mathcal{A}}^{upriv}\,(k) = \ |pr\,(b' = b) - pr\,(random\ coin\ flip)|$$
$$= \left|pr\,(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \ \gg \epsilon$$

Proof: According to the updating procedure of *Yoon*'s protocol $K_i^{T_0} \leftarrow PRNG\left(K_{i-1}^{T_0}\right)$. As a result, following equations can be written

$If\ \ T_b = T_0,$

$$E_{i-1}^{T_b} \oplus D_{i-1}^{T_b} = N_{T,i-1}^{T_b} \oplus PRNG\left(K_{i-1}^{T_b} \oplus C_{i-1}^{T_b}\right) \oplus N_{T,i-1}^{T_b} \oplus K_{i-1}^{T_b}$$
$$= K_{i-1}^{T_0} \oplus PRNG\left(K_{i-1}^{T_0} \oplus C_{i-1}^{T_b}\right)$$

that results in $Adv_{\mathcal{A}}^{upriv}\,(K) = \frac{1}{2} \gg \epsilon$ which means that the target tag can be traceable.

3.4 Forward Traceability Attack

In an RFID authentication protocol this is very important that if an attacker corrupts the secret keys of a specific tag, he/she cannot track the location of the tag in the next sessions. This concept is named forward untraceability. In this section, it is shown that this property is not provided in *Yoon*'s protocol and his protocol suffers from forward traceability attack. In this attack, the attacker uses the fact that the value of EPC_s is fixed in all rounds. To this aim, we show that the attacker can track a specific tag by performing following operations.

Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $(K_i^{T_0}, C_i^{T_0},\ EPC_{s,i}^{T_0})$ from tag T_0. It also sends an $Execute\ query\,(R,\, T_0, i)$ and obtains $N_{R,i}$.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query\,(\,T_0,\ T_1, i)$. 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 $(i+2)$th, the attacker $\mathcal{A}$ sends an $Execute\ query\,(R,\, T_b, i+2)$ by sending $N_{R,i}$ and obtains $\left(M_{1,i+2}^{T_b},\ D_{i+2}^{T_b}\right)$. Now the attacker can compute K_{i+2} at the session $i+2$ by two times repeating $PRNG$ of K_i. Consequently, $N_{T,i+2}$ can be achieved by XORing K_{i+2} and D_{i+2} as $N_{T,i+2} = K_{i+2} \oplus D_{i+2}$, if we have D_{i+2}.

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', first the attacker $\mathcal{A}$ computes $\theta = \ PRNG\left(PRNG\left(K_i^{T_0}\right)\right)$, $\zeta = D_{i+2}^{T_b} \oplus \theta$ and $\gamma = PRNG\left(EPC_{s,i}^{T_0} \oplus N_{R,\ i} \oplus \zeta\right)$, where γ is a 16-bit string. Then, the attacker $\mathcal{A}$ outputs a bit $b' \in \{0,\ 1\}$ as a guess of bit b using the following rule.

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

As a result, it can be written that,

$$Adv_{A}^{upriv}\,(K) = |pr\,(b' = b) - pr\,(random\ coin\ flip)|$$

$$= \left|pr\,(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \epsilon$$

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

$If\ \ T_b = T_0$

$$\begin{aligned} K_{i+2}^{T_b} &= PRNG\left(PRNG\left(K_i^{T_b}\right)\right) && (1)\\ &= PRNG\left(PRNG\left(K_i^{T_0}\right)\right)\\ &= K_{i+2}^{T_0} = \theta\\ N_{T,i+2}^{T_b} &= D_{i+2}^{T_b} \oplus K_{i+2}^{T_b} && (2)\\ &= D_{i+2}^{T_b} \oplus \theta = \zeta\\ (1),(2) &\Longrightarrow && (3)\\ M_{1,i+2}^{T_b} &= K_{i+2}^{T_b} \oplus\\ &\quad \mathrm{PRNG}\left(EPC_{s,i+2}^{T_b} \oplus N_{R,i} \oplus N_{T,i+2}^{T_b}\right) && (4)\\ &= \theta \oplus PRNG\left(EPC_{s,i}^{T_0} \oplus N_{R,i} \oplus \zeta\right)\\ &= \theta \oplus \gamma \end{aligned}$$

4 Analyses of Jung *et al.*'s Protocol

In this part, we analyze the security and privacy of Jung *et al.*'s [40] protocol. We present our privacy analysis based on *Ouafi* and *Phan* privacy model. It is shown that their protocol is vulnerable to DoS attack and also it cannot provide privacy of RFID users. Before presenting our analysis, we have a look at Jung's protocol and explain its steps with more details.

4.1 Jung *et al.*'s Protocol

Jung *et al.*'s protocol is a HMAC-based RFID authentication protocol which is proposed in [40]. This protocol is a mutual authentication protocol which both the tag and the back-end server authenticate each other. The tag and the reader exchange messages over an insecure channel which can be accessed by an attacker. Figure 3 illustrates the authentication procedure of Jung *et al.*'s protocol. As it can be seen, each successful run of this protocol consists of five steps which are given in the rest of this subsection. The notations of Jung *et al.*'s protocol can be found in Table 2.

Step 0: Enrollment phase

1. A random number $(\mathrm{C_0})$, HMAC function, a secret key k, and the tag identifiers $(\mathrm{ID_t})$ have been shared between the tag and the back-end server.
2. Then, a pair $\langle \mathrm{ID_t}, \mathrm{ID_t} \oplus \mathrm{C_0} \rangle$ has been saved in the database of the tag and the back-end server.

Step 1: The reader transmits "Hello" message to the tag with his/her ID $(\mathrm{ID_r})$.

Step 2: Response of the tag

1. The tag selects a random number $(\mathrm{C_1})$
2. Then, the tag computes $\mathrm{ID_t} \oplus \mathrm{C_0}$, $\mathrm{k} \oplus \mathrm{C_0} \oplus \mathrm{C_1}$, $\mathrm{ID_r}$, $\mathrm{T_t}$, and a $= HMAC_{ID_t}\,(\mathrm{T_t}, \mathrm{ID_r})$, and sends them to the reader.

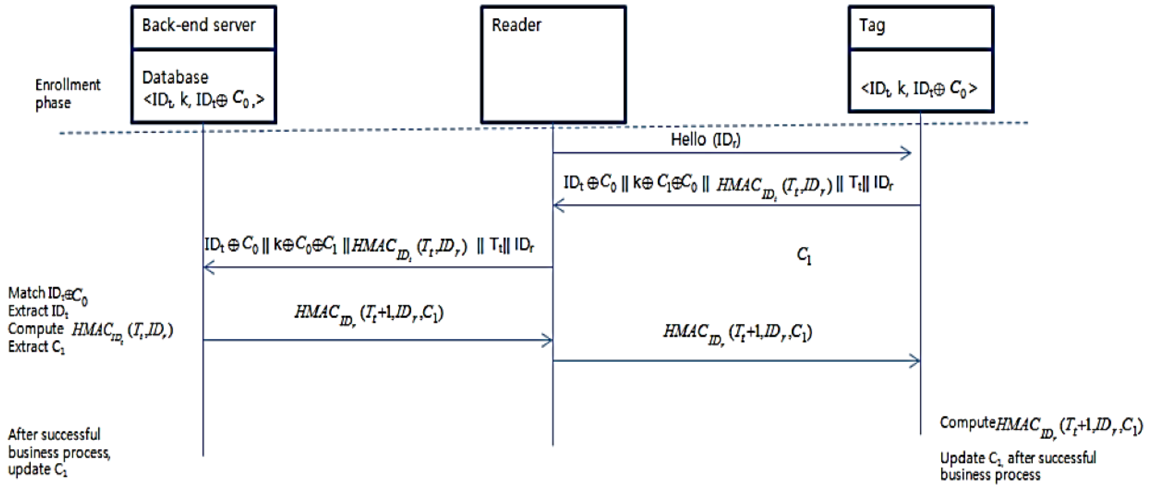


Figure 3. The Jung *et al.*'s Protocol [40].

Table 2. The Notations of Jung *et al.*'s Protocol.

Notation	Description
HMAC	Hash-based Message Authentication Code
C_A	A random number of entity A
C_{new}	A random number of current stage
C_{old}	A random number of previous stage
ID_A	Identity of an entity A
T_A	Timestamp from an entity A
$H(\cdot)$	Hash function
K_i	The authentication key stored in the tag to be used by database to authenticate the tag at the $(i+1)^{th}$ authentication phase
$\|\|$	Concatenation operator

Step 3: The tag authentication

(1) In this step, firstly the reader sends $ID_t \oplus C_0$, $k \oplus C_0 \oplus C_1$, a, ID_r, and T_t to the back-end server.

(2) Secondly, the back-end server matches $ID_t \oplus C_0$ that is in its database with the first part of the received message and obtains $\langle ID_t, k, ID_t \oplus C_0 \rangle$ with $ID_t \oplus C_0$ and uses them to extract ID_t.

(3) After that, the back-end server calculates $a' = HMAC_{ID_t}(T_t, ID_r)$ and $C_1 = k \oplus C_0 \oplus C_1 \oplus k \oplus C_0$.

(4) Then, the back-end server verifies that $a' \overset{?}{=} a$. If the answer is No, it aborts the rest of the protocol.

(5) Next, $\beta = HMAC_{ID_t}(T_t{+}1, ID_r, C_1)$ is calculated by the back-end server and is sent to the reader.

(6) Finally, β is sent to the tag by the reader.

Step 4: The back-end server authentication

(1) In this step, firstly, $\beta' = HMAC_{ID_t}(T_t{+}1, ID_r, C_1)$ is calculated by the tag using his/her T_t, C_1 and received ID_r.

(2) The tag checks that $\beta' = \beta$ or $\beta' \neq \beta$. If $\beta' = \beta$, then the authentication of the back-end server will be confirmed by the tag.

Step 5: Update C_1

After successful authentication in the tag and back-end server, the tag and the back-end server substitute $\langle ID_t, k, ID_t \oplus C_0 \rangle$ with $\langle ID_t, k, ID_t \oplus C_1 \rangle$ that in the next session $ID_t \oplus C_1$ will be used.

4.2 DoS attack on Jung *et al.*'s Protocol

Here,we show that in Jung *et al.*'s protocol, an attacker can make desynchronization between the tag and the back-end server. To this aim, after running four steps of the protocol, when the reader wants to send a message to the tag, the attacker intercepts this transmitted message and stops the protocol. As a result, the back-end server updates $\langle ID_t, k, ID_t \oplus C_0 \rangle$ with $\langle ID_t, k, ID_t \oplus C_1 \rangle$ but the tag does not update its information. As a result, the tag and the back-end server update their secret keys with different values which makes desynchronization between them in the future runs; consequently, in the next runs, the back-end server cannot authenticate the tag.

4.3 Traceability Attack

As mentioned before, providing untraceable and confidential communication is one of the main goals of an RFID authentication protocol. In this section, we show that Jung *et al.* do not provide this property in their protocol. In fact, an attacker can track a specific tag and perform traceability attack against the tag. According to Figure 3, we can see that the ID_t is fixed in all rounds which make the attacker able to perform traceability attack against Jung *et al.*'s protocol as follows,

Learning phase: In round (i), the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$ to the tag by sending $Hello$ message, and obtains $ID^{T_0}_{t,i} \oplus C^{T_0}_i$.

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 $b \in \{0,\ 1\}$, the attacker is given a tag $T_b \in \{T_0,\ T_1\}$. After that, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+1)$ by sending $Hello$ message, and obtains $ID^{T_b}_{t,i+1} \oplus C^{T_b}_{i+1}$.

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

$$b' = \begin{cases} 0 \ \ if\ ID^{T_b}_{t,i+1} \oplus C^{T_b}_{i+1} = ID^{T_0}_{t,i} \oplus C^{T_0}_i \\ 1 \ \ otherwise \end{cases}$$

As a result, it can be written:

$$Adv^{upriv}_{A}(K) = |pr\,(b' = b) - pr\,(random\ coin\ flip)|$$
$$= \left|pr\,(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \epsilon.$$

Proof: After an unsuccessful challenge between the attacker and the tag, the tag does not update $ID^{T_0}_{t,i} \oplus C^{T_0}_i$. Therefore, the tag uses the same value in the next run.

4.4 Backward Traceability Attack

Beside the presented traceability attack in the last subsection, we show that Jung *et al.*'s protocol has another weakness which makes it vulnerable to backward traceability attack. In Jung *et al.*'s protocol, both the secret keys ID_t and k do not update after each successful authentication and they are fixed in all rounds. In the rest of this subsection, it can be seen that how an attacker can use this fact as a privacy flaw and he/she performs backward traceability attack against Jung *et al.*'s protocol.

Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $K^{T_0}_i$ from tag T_0. After that, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$, and obtains $\alpha_i = ID^{T_0}_{t,i} \oplus C^{T_0}_i$.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query\,(\,T_0,\ T_1, i)$. 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 $(i-1)$th, the attacker $\mathcal{A}$ sends an $Execute\ query\,(R,\ T_b, i-1)$, and obtains $\alpha_{i-1} = ID^{T_b}_{t,i-1} \oplus C^{T_b}_{i-1}$ and $\beta_{i-1} = k^{T_b}_{i-1} \oplus C^{T_b}_i \oplus C^{T_b}_{i-1}$.

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 determine $b' \in \{0,\ 1\}$, the attacker uses the following rule.

$$b' = \begin{cases} 0\ if\ \ \alpha_{i-1} \oplus \beta_{i-1} = \alpha_i \oplus k^{T_0}_i \\ 1\ otherwise \end{cases}$$

As a result, it can be written:

$$Adv^{upriv}_{A}(k) = |pr\,(b' = b) - pr\,(random\ coin\ flip)|$$
$$= \left|pr\,(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \epsilon$$

Proof: Since the value of ID_t and k are fixed in all rounds, then $k^{T_0}_i = k^{T_0}_{i-1}$ and $ID^{T_0}_{t,\,i} = ID^{T_0}_{t,\,i-1}$. Using this fact, the following equations can be written.

$If\ \ T_b = T_0$

$$\alpha_{i-1} \oplus \beta_{i-1} = ID^{T_b}_{t,i-1} \oplus C^{T_b}_{i-1} \oplus k^{T_b}_{i-1} \oplus C^{T_b}_i \oplus C^{T_b}_{i-1}$$
$$= ID^{T_b}_{t,i-1} \oplus k^{T_b}_{i-1} \oplus C^{T_b}_i$$
$$= ID^{T_0}_{t,i} \oplus k^{T_0}_i \oplus C^{T_0}_i = \alpha_i \oplus k^{T_0}_i$$

5 Improved Protocols

In Section 3 and 4, it is shown that both the *Yoon* and Jung *et al.*'s protocols have some drawbacks and cannot provide secure and untraceable authentication for RFID end-users. In this Section, in order to overcome all the reported weaknesses on *Yoon* and Jung *et al.*'s protocol, we propose some modifications on their structures and propose an improved version of each one.

5.1 Improvements on Yoon's Protocol

In Section 3, we observed that in the structure of *Yoon*'s protocol there are two major problems in updating C_i and K_i that make the protocol vulnerable to various traceability attacks. In order to prevent these attacks and increase the privacy of this protocol, we change the way of updating C_i and K_i as follows,

$$C_{i+1} \leftarrow PRNG\,(N_T \oplus N_R \oplus P_i)$$
$$K_{i+1} \leftarrow PRNG\,(K_i \oplus N_3)$$

where N_3 is a new random number that is generated in the tag. Furthermore, some changes are applied in the tag's processes and authentication procedure in the back-end server. Figure 4 shows the improved version of *Yoon*'s protocol which can be summarized as follows,

a) Initial phase

Similar to the *Yoon*'s protocol, some initial secret values such as K_0, P_0 and C_0 that are generated randomly in the manufacture, and these are shared between the tag and the back-end server. Also, the corresponding values of the mentioned parameters in the back-end server are set to these initial values ($K_{old} = K_{new} = K_0$, $P_{old} = P_{new} = P_0$ and $C_{old} = C_{new} = C_0$).

b) Authentication phase

This phase includes five steps as follows,

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

Step 2. Tag → Reader: Upon receiving N_R, the tag generates random numbers N_T and N_3. Then it computes the following messages and sends them along with C_i to the reader.

$$\begin{aligned} M_1 &= PRNG\left(EPC_s \oplus N_R \oplus N_T\right) \oplus K_i, \\ D &= N_T \oplus K_i, \\ C_i &= C_i \oplus N_3, \\ E &= PRNG(N_T) \oplus PRNG(C_i \oplus K_i). \end{aligned}$$

Step 3. Reader → Back-end server: The reader calculates $V = H(RID \oplus N_R)$ and forwards the messages $(M_1, D, C_i, E,\ N_R, V)$ to the back-end server.

Step 4. Back-end server → Reader: Based on the received messages from the reader, the back-end server performs the following operations,

(1) The back-end server verifies $V \overset{?}{=} H(RID \oplus N_R)$ and follows the rest of authentication procedure.

(2) The back-end server first computes $I_X = M_1 \oplus K_X$ for $X \in \{old,\ new\}$. Then it checks whether $I_X = PRNG(EPC_s \oplus N_R \oplus D \oplus K_X)$ and determines that $X = old$ or new.

(3) Now using the obtained $X = old$ or new, the back-end server verifies $E \overset{?}{=} PRNG(N_T) \oplus PRNG(C_i \oplus K_X)$. If $E = PRNG(N_T) \oplus PRNG(C_i \oplus K_X)$, it authenticates the tag and responds to the reader by the following messages,

$$\begin{aligned} M_2 &= PRNG(EPC_s \oplus N_T) \oplus P_X \\ Info &= DATA \oplus RID \\ MAC &= H(DATA \oplus N_R), \end{aligned}$$

otherwise, the back-end server aborts the protocol.

(4) Finally, the back-end server computes $N_3 = C_i \oplus C_X$ and updates its secret values as follows,

$$\begin{aligned} &If \quad X = new \\ &\quad K_{old} \leftarrow K_{new} \leftarrow PRNG\left(K_{new} \oplus N_3\right) \\ &\quad C_{old} \leftarrow C_{new} \leftarrow PRNG\left(N_T \oplus N_R \oplus P_X\right) \\ &\quad P_{old} \leftarrow P_{new} \leftarrow PRNG\left(P_{new}\right) \\ &Else \\ &\quad C_{new} \leftarrow PRNG\left(N_T \oplus N_R \oplus P_X\right) \\ &End \end{aligned}$$

Step 5. Reader → Tag: Now using the received message $Info$, the reader computes $DATA = Info \oplus RID$, and verifies $H\left(DATA \oplus N_R\right) \overset{?}{=} MAC$. If the verification is successful, the reader sends M_2 to the tag.

Finally utilizing the received message M_2, the tag verifies $M_2 \oplus P_X \overset{?}{=} PRNG(EPC_s \oplus N_T)$. If the answer is Yes, the tag updates its secret values by,

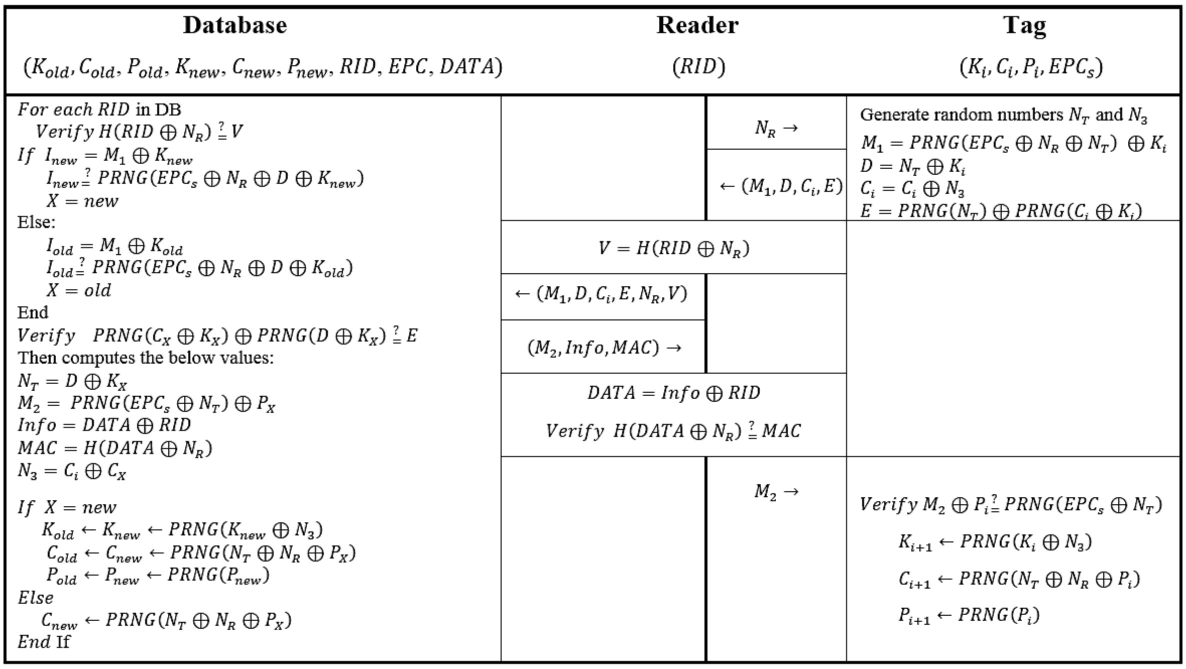
$$\begin{aligned} K_{i+1} &\leftarrow PRNG\left(K_i \oplus N_3\right) \\ C_{i+1} &\leftarrow PRNG\left(N_T \oplus N_R \oplus P_i\right), \\ P_{i+1} &\leftarrow PRNG\left(P_i\right) \end{aligned}$$

otherwise, the tag aborts the protocol.

In the rest of this section, some analyses are presented and it is shown that how new changes make the improved protocol resistant against different traceability attacks.

- **Traceability Attack**

In [34] and [35] Safkhani *et al.* and Mohammadali *et al.* respectively, presented two individual traceability attacks against *Yoon*'s protocol [20] which both are based on ad-hoc methods. Besides, in Section 3.2 we formally showed that in *Yoon*'s protocol, the structure of $C_i = PRNG\left(N_T \oplus N_R\right)$ has some problems that makes it vulnerable against traceability attack. In the improved protocol, in order to prevent this attack, we have replaced generating $E = N_T \oplus PRNG\left(C_i \oplus K_i\right)$ with $E = PRNG\left(N_T\right) \oplus PRNG\left(C_i \oplus K_i\right)$. Also, we have modified the structure of the transmitted C_i as $C_i = C_i \oplus N_3$, where N_3 is a new random number that is generated by the tag. Note that with the first modification, the dependency between the E and D is omitted and an attacker cannot trace the tag by



Database

$(K_{old}, C_{old}, P_{old}, K_{new}, C_{new}, P_{new}, RID, EPC, DATA)$

For each RID in DB
  $Verify\ H(RID \oplus N_R) \overset{?}{=} V$
$If\ \ I_{new} = M_1 \oplus K_{new}$
    $I_{new} \overset{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{new})$
    $X = new$
Else:
    $I_{old} = M_1 \oplus K_{old}$
    $I_{old} \overset{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{old})$
    $X = old$
End
$Verify\ \ PRNG(C_X \oplus K_X) \oplus PRNG(D \oplus K_X) \overset{?}{=} E$
Then computes the below values:
$N_T = D \oplus K_X$
$M_2 = PRNG(EPC_s \oplus N_T) \oplus P_X$
$Info = DATA \oplus RID$
$MAC = H(DATA \oplus N_R)$
$N_3 = C_i \oplus C_X$

$If\ \ X = new$
    $K_{old} \leftarrow K_{new} \leftarrow PRNG(K_{new} \oplus N_3)$
    $C_{old} \leftarrow C_{new} \leftarrow PRNG(N_T \oplus N_R \oplus P_X)$
    $P_{old} \leftarrow P_{new} \leftarrow PRNG(P_{new})$
$Else$
    $C_{new} \leftarrow PRNG(N_T \oplus N_R \oplus P_X)$
End If

Reader

(RID)

$N_R \rightarrow$

$\leftarrow (M_1, D, C_i, E)$

$V = H(RID \oplus N_R)$

$\leftarrow (M_1, D, C_i, E, N_R, V)$

$(M_2, Info, MAC) \rightarrow$

$DATA = Info \oplus RID$

$Verify\ \ H(DATA \oplus N_R) \overset{?}{=} MAC$

$M_2 \rightarrow$

Tag

(K_i, C_i, P_i, EPC_s)

Generate random numbers N_T and N_3
$M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$
$D = N_T \oplus K_i$
$C_i = C_i \oplus N_3$
$E = PRNG(N_T) \oplus PRNG(C_i \oplus K_i)$

$Verify\ M_2 \oplus P_i \overset{?}{=} PRNG(EPC_s \oplus N_T)$
    $K_{i+1} \leftarrow PRNG(K_i \oplus N_3)$
    $C_{i+1} \leftarrow PRNG(N_T \oplus N_R \oplus P_i)$
    $P_{i+1} \leftarrow PRNG(P_i)$

Figure 4. Improved Version of Yoon's Protocol.

XORing them. Moreover, by applying the second modification, the value of C_i is changed in each run of protocol and an attacker cannot trace the tag even if the tag does not update its secret values.

- **Backward and Forward Traceability Attacks**

In Section 3, we have observed that the privacy of *Yoon*'s protocol has some problems that makes it vulnerable against backward and forward traceability attacks. In the proposed protocol, in order to enhance the privacy and remove all mentioned privacy attacks, we apply two changes in the updating procedures. More precisely, we have changed the way of updating $C_i = PRNG\,(N_T \oplus N_R)$ and $K_i = PRNG\,(K_i)$ with $C_i = PRNG\,(N_T \oplus N_R \oplus P_i)$ and $K_i = PRNG\,(K_i \oplus N_3)$, respectively where N_3 is a new random number that is generated by the tag. As it can be seen, by applying these changes if an attacker obtains the secret values K_i and C_i, it cannot perform backward and forward traceability attacks. As a result, the proposed protocol is secure against two mentioned privacy attacks.

- **DoS Attack**

Besides the mentioned analyses, the proposed protocol is secure against DoS attack. In this attack, the attacker tries to create desynchronization between the tag and the back-end server. The attacker can perform this attack through three different methods. First, it can intercept the last step of authentication phase between the back-end server and the tag and desynchronizes them in the next runs. In the second and the third methods, first the attacker needs to perform tag impersonation and reader impersonation attacks. After performing these attacks, it can perform DoS attack and desynchronize the tag and the back-end server by two different methods similar to [43].

Since in the improved protocol, both the old and the new secret keys are stored in the back-end server, the attacker cannot perform DoS attack by intercepting. Moreover, in the improved protocol by applying a change in the tag's response E, the protocol has become secure against the impersonation attack. As a result, in the proposed protocol the attacker cannot desynchronize the tag and the back-end server similar to the presented attacks in [43].

5.2 Improvements on Jung *et al.*'s Protocol

According to the presented analysis in Section 4, it is shown that Jung *et al.*'s protocol suffers from DoS attack, traceability attack and backward traceability attack. In order to overcome all the mentioned weaknesses, we propose some modifications in the way of updating the secret values, the structure of response messages from the tag, and the stored data in the back-end server and the tag. The modified version of Jung *et al.*'s protocol consists of five steps as follows,

Step 0: Enrollment phase

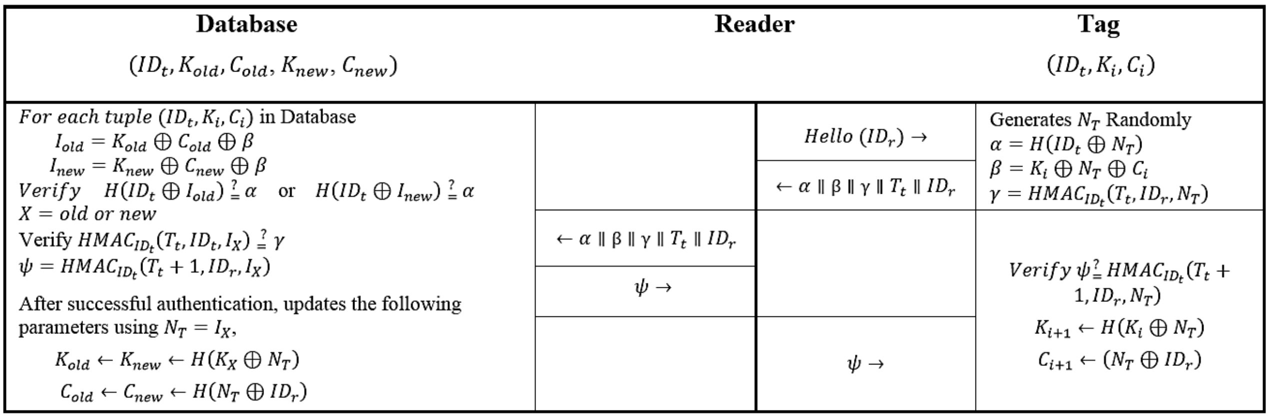


Figure 5. Improved Version of Jung *et al.*'s Protocol.

(1) A random number (C_0), HMAC function, a secret key K_i, and the tag identifiers (ID_t) are shared between the tag and the back-end server.
(2) Then, parameters $\langle ID_t, K_{old}, K_{new}, C_{old}, C_{new} \rangle$ are saved in the database of the back-end server and parameters $\langle ID_t, K_i, C_i \rangle$ are saved in the tag.

Step 1: The reader transmits "Hello" message to the tag with his/her ID (ID_r).

Step 2: Response of the tag

(1) The tag computes a random number N_T.
(2) Then, the tag computes following messages and sends them along with T_t and ID_r to the reader.

$$\alpha = H(ID_t \oplus N_T)$$
$$\beta = K_i \oplus N_T \oplus C_i$$
$$\gamma = HMAC_{ID_t}(T_t, ID_r, N_T)$$

Step 3: The tag authentication

(1) The reader sends the received messages from the tag to the back-end server.
(2) The back-end server computes $I_X = K_X \oplus C_X \oplus \beta$ for each tuple of $\langle ID_t, K_X, C_X \rangle$, where $X \in \{old,\ new\}$. Then, in order to determine X, the back-end server verifies $H(ID_t \oplus I_X) \overset{?}{=} \alpha$.
(3) Now the back-end server authenticates the tag by verifying $HMAC_{ID_t}(T_t, ID_r, I_X) \overset{?}{=} \gamma$.
(4) Then the back-end server calculates the message $\Psi = HMAC_{ID_t}(T_t + 1, ID_r, I_X)$ and sends it to the tag through the reader.

Step 4: The back-end server authentication

(1) In this step, firstly the tag generates N_T and then uses received ID_r and his/her T_t and calculates, $\Psi' = HMAC_{ID_t}(T_t + 1, ID_r, N_T)$ is calculated by the tag using his/her T_t, N_T and received ID_r.
(2) The tag checks whether $\Psi' = \Psi$ or not. If the answer is Yes, then the authentication of the back-end server will be confirmed by the tag.

Step 5: Updating phase

After successful authentication in the tag and back-end server, they update their secret parameters as follows,

(1) The back-end server updates as follows,

$$K_{old} \leftarrow K_{new} \leftarrow H(K_X \oplus N_T)$$
$$C_{old} \leftarrow C_{new} \leftarrow H(N_T \oplus ID_r).$$

(2) The tag updates as follows,

$$K_{i+1} \leftarrow H(K_i \oplus N_T)$$
$$C_{i+1} \leftarrow (N_T \oplus ID_r).$$

Figure 5 illustrates the structure of the improved version of Jung *et al.*'s protocol. The reasons of the main changes can be expressed as follows,

- In order to prevent DoS attack, both the new and old secret values are saved in the back-end server. In this case, if an attacker intercepts the protocol and prevents updating the secret values, since the back-end server saves the current and the previous secret values, the proposed protocol is not vulnerable to DoS attack.
- In order to prevent traceability attack we have applied a change in the tag's responses as follows,

$$\alpha = H(ID_t \oplus N_T)$$

where N_T is a random number that is generated by the tag. It is worth to mention that by using a hash function and a random number N_T in generating α, the attacker cannot perform traceability attack against the improved Jung *et al.*'s protocol, even if he/she intercepts the protocol.

- Finally, in order to prevent backward traceability attack we update K_i and C_i in the tag and the back-end server as follows,

$$K_{old} \leftarrow K_{new} \leftarrow H\left(K_X \oplus N_T\right)$$

$$C_{old} \leftarrow C_{new} \leftarrow H\left(N_T \oplus ID_r\right).$$

With these changes, it can be seen that if the attacker obtains K_i and C_i, it cannot calculate K_{i-1} and C_{i-1} to perform the backward traceability attack.

In Table 3, the security and the privacy of the proposed protocols are compared with analyzed protocols. According to the analysis, it can be seen that the proposed protocols are resistant against the mentioned attacks. It can be conclude that the improved protocols can protect RFID users against various security and privacy threats.

Table 3. Analyses of the Proposed Protocols.

Protocol Notation	Yoon [20]	Jung *et al.* [40]	Improved Yoon	Improved Jung *et al.*
DoS Attack	×	×	✓	✓
Traceability Attack	×	×	✓	✓
Backward Traceability	×	×	✓	✓
Forward Traceability	×	✓	✓	✓

✓: Secure ×: Insecure

6 Conclusion

We have analyzed the privacy of two recent lightweight RFID authentication protocols that have been proposed by *Yoon* and Jung *et al.* We have shown that both protocols have some flaws and are vulnerable against various attacks. We showed that *Yoon*'s protocol is not secure against all types of traceability attacks including *traceability* attack, *backward traceability* and *forward traceability* attacks. Also, we have shown that Jung *et al.*'s protocol cannot provide security and privacy of RFID users and it is vulnerable against DoS attack, traceability and backward traceability attacks. In addition, in order to safeguard the investigated protocols, we have proposed a modified version of each one. Our analyses show that improved protocols overcome all the reported problems and prevent the presented attacks. As a result, the proposed protocols can be successful schemes for providing privacy of RFID users in different identification and authentication applications.

Acknowledgment

This work was partially supported by Iran NSF (INSF) under grant number 92.32575.

References

[1] D. Heyden, "RFID Applications," Available: http://www.fibre2fashion.com/industry-article/11/1023/ rfid-applications1.asp.

[2] S. Maharjan, "RFID and IOT: An overview," Simula Research Laboratory University of Oslo, 2010.

[3] L. Yang, P. Yu, W. Bailing, Q. Yun, B. Xuefeng, and Y. Xinling, "Hash-based RFID Mutual Authentication Protocol," *International Journal of Security & Its Applications,* vol. 7, no. 3, pp. 1738-9976, 2013.

[4] B. Song and C. J. Mitchell, "Scalable rfid security protocols supporting tag ownership transfer," *Comput. Commun.,* vol. 34, pp. 556-566, 2011.

[5] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 2, p. 381–394, 2006.

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

[7] B. Alomair, A. Clark, J. Cuellar, and 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.

[8] K. Ouafi, "Security and privacy in RFID systems," PhD Thesis, Ecole Polytechnique Federale DE Lausanne, 2008.

[9] M. R. Alagheband, and M. R. Aref, "Simulation-based traceability analysis of RFID authentication protocols," *Wireless Personal Communications,* vol. 77, no. 2, pp. 1020-1038, 2014.

[10] B. Hameed, I. Khan, F. Durr, and K. Rothermel, "An RFID based consistency management framework for production monitoring in a smart real-time factory," in *2nd International Conference on the Internet of Things (IoT),* Tokyo, 2010.

[11] D. He, and Sh. Zeadally, "An analysis of RFID authentication schemes for Internet of things in healthcare environment using elliptic curve cryptography," *IEEE Internet of Things Journal,* vol. 2, no. 1, pp. 72 - 83, 2015.

[12] G. Avoine and X. Carpent, "Yet another ultralightweight authentication protocol that is broken," in *Workshop on RFID Security - RFIDSec'12,* Nijmegen, 2012.

[13] M. Asadpour, and M. T. Dashti, "A privacy-friendly RFID protocol using reusable anonymous tickets," in *10th International Conference on Trust, Security and Privacy in Computing and Communications,* Changsha , 2011.

[14] Z. Sohrabi-Bonab, M. Alagheband, and M. R. Aref, "Traceability analysis of quadratic residue-based RFID authentication protocols," in *Eleventh Annual International Conference on Privacy, Security and Trust (PST),* Tarragona , 2013.

[15] M. R. Alagheband, and M. R. Aref, "Unified

privacy analysis of new founded RFID authentication protocols," *Security and Communication Networks,* vol. 6, no. 8, pp. 999-1009, 2013.

[16] M. H. Habibi, M. R. Aref, and Di Ma, "Addressing flaws in RFID authentication protocols," *Progress in Cryptology, INDOCRYPT 2011, LNCS 7107,* vol. 7, p. 216–235 , 2011.

[17] P. Babvey, H. A. Yajam, and T. Eghlidos, "Security analysis of SKI protocol," in *11th International ISC Conference on Information Security and Cryptology (ISCISC),* Tehran, 2014.

[18] "EPCglobal Inc.," Available: http://www.epcglobalinc.org.

[19] H. Y. Chien, and C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," *Computer Standards & Interfaces,* vol. 29, no. 2, pp. 254-259, 2007.

[20] 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.

[21] 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.

[22] T. C. Yeh, Y. J. Wanga, T. Ch. Kuo, and S. S. Wanga, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard," *Expert Systems with Applications,* vol. 37, p. 7678–7683, 2010.

[23] F. Xiao, Y. Zhou, J. Zhou, H. Zhu, and X. Niu, "Security protocol for RFID system conforming to EPC-C1G2 standard," *Journal of Computers,* vol. 8, no. 3, pp. 605-612, 2013.

[24] M. Safkhani, N. Bagheri, P. Peris-Lopez, A. Mitrokotsa, J. C Hernandez-Castro, "Weaknesses in another Gen2-based RFID authentication protocol," in *IEEE International Conference on RFID-Technologies and Applications (RFID-TA),* 2012.

[25] D. N. Duc, J. Park, H. Lee, and K. Kim, " Enhancing security of EPC global Gen-2 RFID tag against traceability and cloning," in *Symposium on Cryptography and Information Security (CSIS),* pp. 17-20, 2006.

[26] S. Karthikeyan, and M. Nesterenko, "RFID security without extensive cryptography," in *3rd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN),* pp. 63–67, 2005.

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

[28] I. Coisel, and T. Martin, "Untangling RFID privacy models," *Journal of Computer Networks and Communications,* pp. 1-26, 2013, doi:10.1155/2013/710275.

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

[30] 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.

[31] 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.

[32] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for RFID privacy," in *15th European Symposium on Research in Computer Security (ESORICS),* Athens, 2010.

[33] D. Moriyama, S. Matsuo, and M. Ohkubo, "Relation among the security models for RFID authentication," in *17th European symposium on research in computer security (ESORICS),* pp. 661–678, 2012.

[34] M. Safkhani, N. Bagheri, S. K. Sanadhya, and M. Naderi, "Cryptanalysis of improved Yeh et al. 's authentication Protocol: An EPC Class-1 Generation-2 standard compliant protocol," http://eprint.iacr.org/2011/426.pdf, 2011.

[35] A. Mohammadali, Z. Ahmadian, and M. R. Aref, "Analysis and Improvement of the securing RFID systems conforming to EPC Class 1 Generation 2 standard," *IACR Cryptology ePrint Archive,* vol. 66, pp. 1-9, 2013.

[36] K. Baghery, B. Abdolmaleki, B. Akhbari, and M. R. Aref, "Privacy analysis and improvements of two recent RFID authentication protocols," in *11th International ISC Conference on Information Security and Cryptology (ISCISC),* Tehran, 2014.

[37] S.-P. Wang, Q.-M. Ma, Y.- L. Zhang, and Y.-S. Li, "A HMAC-Based RFID Authentication Protocol," in *2nd International Symposium on Information Engineering and Electronic Commerce (IEEC),* 2010.

[38] 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 Communication,* vol. 34, pp. 391-397, 2011.

[39] J. Cho, S-C. Kim, and S. K. Kim, "Hash-based RFID tag mutual authentication scheme with retrieval efficiency," in *9th IEEE Internation Symposium on Parallel and Distributed Processing with Applications,* 2011.

[40] S. W. Jung, and S. Jung, "HMAC-based RFID authentication protocol with minimal retrieval at server," *The Fifth International Conference*

on Evolving Internet, pp. 52-55, 2013.

[41] Y. C. Huang, and J. R. Jiang, "Ultralightweight RFID reader-tag mutual authentication revisited," in *IEEE International Conference on Mobile Services (MS),* New York, 2015.

[42] D. Z. Sun, and J. D. Zhong, "A hash-based RFID security protocol for strong privacy protection," *IEEE Transactions on Consumer Electronics,* vol. 58, no. 4, pp. 1246-1252, 2012.

[43] B. Abdolmaleki, K. Baghery, B. Akhbari, and M. R. Aref, "Attacks and improvements on two newfound RFID authentication protocols," in *7th International Symposium on Telecommunications (IST),* Tehran, 2014.

Karim Baghery is a graduate research assistant at Information Systems and Security Laboratory (ISSL), Sharif University of Technology, Tehran, Iran. He received his M.S. degree in Electrical Engineering (Communications Systems) from Shahed University Tehran, Iran in 2014, and the B.S. degree in Electrical Engineering (Telecommunications) from IAU University, Urmia Branch, Iran, in 2010. During 2012 to 2014 he was working in the Information Theoretic Learning Systems Laboratory (ITLSL), Department of Engineering, Shahed University, Tehran, Iran. He is Member of IEEE since 2013 and he is invited reviewer of KSII Transactions on Internet and Information Systems and Wireless Personal Communications international journals. His research interests mainly include lightweight cryptography, RFID security and privacy, internet of things and optimization on wireless networks.

Behzad Abdolmaleki is a research assistant at Information Systems and Security Laboratory (ISSL), Sharif University of Technology, Tehran, Iran since 2013. He received his M.S. degree in Electrical Engineering-Communications from Shahed University, Iran in 2014 and B.S. degree in physics from university of Kurdistan, Sanandaj, Iran, in 2010. Since 2014, he is Member of IEEE. His research interests include information security, cryptography, E-voting, and cooperative communications.

Bahareh Akhbari received the B.S. degree in 2003, the M.S. degree in 2005 and the Ph.D. degree in 2011 all in Electrical Engineering from Sharif University of Technology (SUT), Tehran, Iran. She was also a visiting Ph.D. student at the University of Minnesota for one year, starting in 2010. Since 2012, she is an assistant professor of the Faculty of Electrical Engineering, K. N. Toosi University of Technology (KNTU), Tehran, Iran. Her research interests include network information theory, communication theory, cryptography and network security.

Mohammad Reza Aref received the B.S. degree in 1975 from University of Tehran, Iran, and the M.S. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in electrical engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a Faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of electrical engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 230 technical papers in communication and information theory and cryptography in international journals and conferences proceedings. His current research interests include areas of communication theory, information theory, and cryptography.