Game-Based Privacy Analysis of RFID Security Schemes for Confident Authentication in IoT

Behzad Abdolmaleki · Karim Baghery · Shahram Khazaei · Mohammad Reza Aref

Abstract Recently, Radio Frequency Identification (RFID) and Near Field Communication (NFC) systems are found in various user-friendly services that all of us deal with in our daily lives. As these systems are ubiquitously deployed in different authentication and identification applications, inferring information about our behavior will be possible by monitoring our use of them. In order to provide privacy and security requirements of RFID users in novel authentication applications, lots of security schemes have been proposed which have tried to provide secure and untraceable communication for end-users. In this paper, we investigate the privacy of three RFID security schemes which have been proposed recently. For privacy analysis, we use the well-known RFID formal privacy model proposed by *Ouafi* and *Phan*. We show that all the studied protocols have some privacy drawbacks, making them vulnerable to various traceability attacks. Moreover, in order to overcome all the reported weaknesses and prevent the presented attacks, we apply some modifications in the structures of the studied protocols and propose an improved version of each one. Our analyses show that the modified protocols are more efficient than their previous versions and new modifications can omit all the existing weaknesses on the analyzed protocols. Finally, we compare the modified protocols with some new-found RFID authentication protocols in the terms of security and privacy.

Keywords RFID authentication protocols · Traceability attacks · Internet of Things · EPC C1 G2 standard · Hash functions

1 Introduction

Radio Frequency Identification (RFID) systems are a popular and prominent strategy for fast and accurate identification and authentication in different domains [1]. These systems use radio waves to automatically capture data for the mentioned purposes. Track with precision, production control, supply chain management, asset management, healthcare control, and pass control are some applications which can be done easily by RFID systems [2-6].

In general, each RFID system consists of a large number of RFID tags, RFID readers and a database. A structure of an RFID system is shown in Fig. 1. An RFID tag consists of an electronic chip and a microstrip antenna which uses them for

✉ B. Abdolmaleki
Information Systems and Security Lab (ISSL), Sharif University of Technology.
e-mail: b.abdolmaleki.ir@ieee.org

K. Baghery
ISSL Lab, Sharif University of Technology.
e-mail: k.baghery.1988@ieee.org

S. Khazaei
Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran.
e-mail: shahram.khazaei@sharif.edu

M. R. Aref
ISSL Lab, Electrical Engineering Department, Sharif University of Technology, Tehran, Iran.
e-mail: aref@sharif.edu

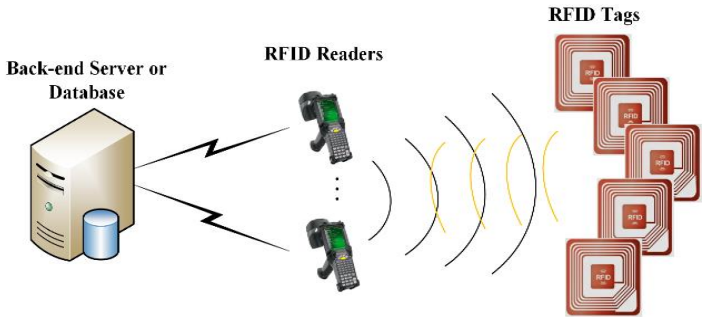

Fig. 1. An RFID system architecture.

connection with the readers. In different applications, based on tag's power, memory, operational frequency, and computational capabilities, various tags can be chosen [7]. Second part of an RFID system is the reader which is located between the tag and the database and in identification and authentication processes exchanges some messages between them. Finally, the main part of each RFID system is the database or back-end server which includes high speed processors and all secret information about tags [8].

Above all, RFID systems are an interesting candidate to implant in the Internet of Thing (IoT) system which is a huge network of IP-based objects which will communicate automatically and without human interposition [9]. In the IoT paradigm, various sensing devices will be deployed to make a connection between objects in our environment. In this paradigm, RFID tags can be attached to various things to communicate with RFID readers which will play the role of an IoT gateway to connect the IoT global network [10]. A communication scenario of RFID tags and readers in the IoT network is shown in Fig. 2. Although, in some cases connections between IoT elements are not important and seem to be trivial, they create some new concerns. In order to avoid these concerns, all connections between the objects and humans need to be secure, confidential and controlled [10]. In addition, an RFID system can be an excellent choice for tracking different objects in different application. Tracking the owner of E-passports, tracking people by the bought products, tracking the readers by the barrowed books, and tracking pets are some of the RFID systems applications with obvious privacy concerns [8]. In order to overcome these concerns and provide RFID end-users security and privacy, a lot of security schemes have been proposed [11-17].

Electronic Product Code Class 1 Generation 2 (EPC C1 G2) standard is one of most popular standards for RFID passive tags which provided by EPCglobal organization [18]. Until now, lots of RFID security schemes have been proposed under EPC C1 G2 standard [16,17]. In [17], *Pang et al.* have proposed an RFID mutual authentication protocol based on EPC C1 G2 standard. They have claimed that their protocol is secure against different attacks and can provides user privacy. However, in [13], *Wang et al.* showed that still *Pang et al.*'s protocol has some weaknesses and it is vulnerable to Denial-of-

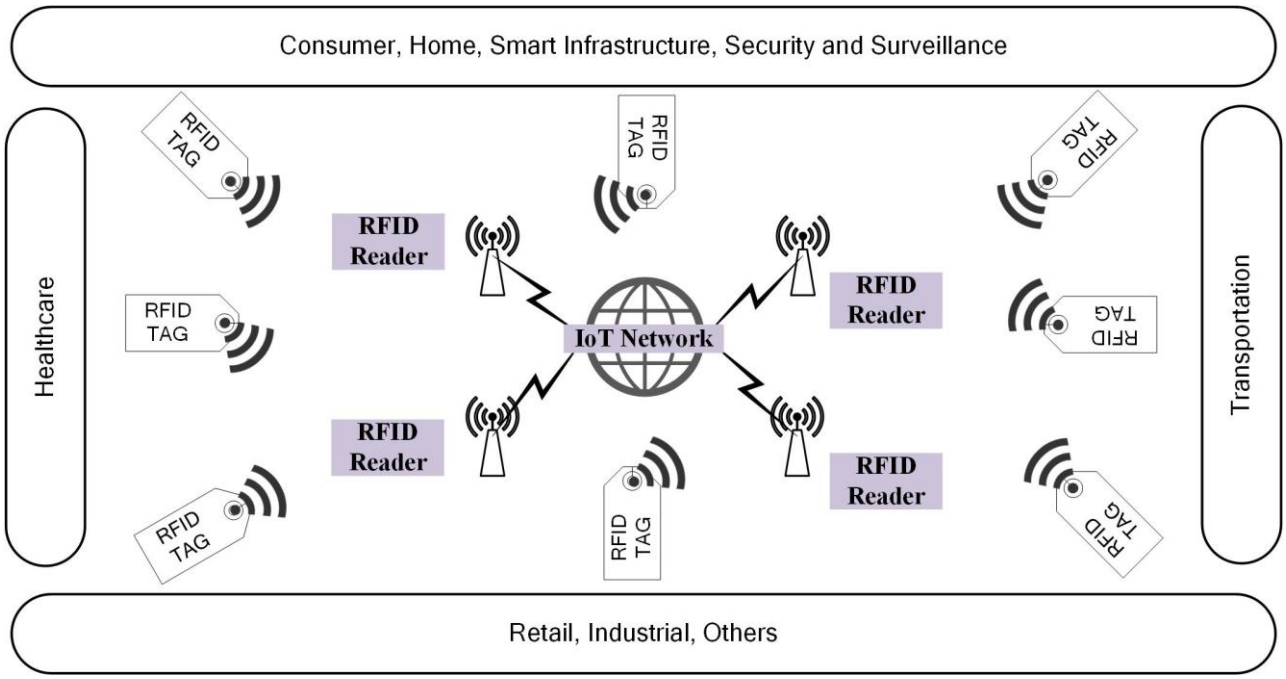

Fig. 2. A communication scenario of RFID tags and readers in the IoT network.

Service (DoS) attack and also the adversary can obtain secret parameters with $O(2^{16})$ attack complexity. Then, they applied some changes on *Pang et al.*'s protocol and proposed an improved version of it. *Wang et al.* analyzed the improved protocol and claimed that it is secure against various security and privacy attacks and an attacker cannot obtain secret keys with $O(2^{16})$ attack complexity. In this paper, we cryptanalyze *Wang et al.*'s protocol and we show that still there are some flaws in their protocol: it suffers from traceability and forward traceability attacks, the problem of secret parameter reveal is not yet solved and an adversary can obtain secret values with $O(2^{16})$ attack complexity.

Other new application of RFID systems are in medicine and healthcare systems [19]. For example, an RFID tag build into an armband could contain a unique identifier for a patient. In [20], *Chen et al.*'s proposed an RFID access control authentication protocol for different authorization mechanism. In their protocol, it is assumed that both communication channels between the tag and the back-end server are insecure and can be eavesdropped by an adversary. *Chen et al.* have tried to provide secure and confidential protocol against various security and privacy attacks. Recently, in [21], *Safkhani et al.* analyzed *Chen et al.*'s protocol and illustrated that it suffers from impersonation (tag, reader, and back-end server), DoS and traceability attacks. Then, in order to overcome all the mentioned weaknesses, *Safkhani et al.* proposed some modifications on *Chen et al.*'s protocol and proposed a strengthened version of it [21]. *Safkhani et al.* have claimed that their strengthened protocol is secure against all types of active and passive attacks. In this study, we show that *Safkhani et al.*'s modifications on *Chen et al.*'s protocol did not overcome all the previous problems and still it has some weaknesses. It is shown that *Safkhani et al.*'s protocol cannot provide users privacy and it suffers from traceability and forward traceability attacks. Then, in order to omit the mentioned weaknesses, we apply some changes on authentication phase of *Safkhani et al.*'s protocol. Our analyses show that with new changes all the existing weaknesses are eliminated and the modified protocol can provide secure and confidential communication for RFID users in different access control applications such as healthcare systems.

Another direction of researchers for designing RFID authentication protocol is using the cryptographic hash functions [22,23]. In 2008, *Ha et al.* [22] proposed a hash-based RFID authentication which protects the exchanged messages with a hash function. In 2012, *Sun* and *Zhong* showed that although *Ha et al.* have tried to protect exchanged messages among the tag and the back-end server, still their protocol does not provide forward privacy [23]. Then, in order to remove this problem, *Sun* and *Zhong* proposed a modification and proposed an improved version of *Ha et al.*'s protocol. *Sun* and *Zhong* claimed that their improved protocol has strong privacy and it is resistant against various traceability attacks. However, we show that *Sun* and *Zhong* were not successful in omitting the mentioned weakness and still their improved protocol cannot provide confidential communication for RFID end-users. More precisely, we show that an attacker can perform forward traceability attack against *Sun* and *Zhong*'s protocol and trace a target tag. In addition, in order to remove this weakness, a modified version of *Sun* and *Zhong*'s protocol is proposed which efficiently uses hash functions to prevent various security and privacy attacks.

Basically, RFID authentication protocols can be analyzed based on *Formal* and *Ad-hoc* methods. In the Ad-hoc methods, an adversary defines some new variations and uses them on his/her analysis. On the other hand, in the Formal methods, the abilities of the adversary defined in different queries and an adversary uses them to perform a specific attack. It is shown that in order to discover all weaknesses of an RFID authentication protocol, using a formal privacy model is necessary [24]. During the last decade, in order to analyze privacy of RFID authentication protocols, several formal privacy models have been presented [25-28]. In 2008, *Ouafi* and *Phan* [28] proposed a game-based formal privacy model which is one of the well-known models for traceability analysis of RFID authentication protocols (referred to as *Ouafi-Phan*). We present our traceability analyses based on *Ouafi-Phan* privacy model.

The paper is organized as follows: We review *Ouafi-Phan* formal privacy model and related attacks in Sect. 2, which are essential concepts in our presented analysis. In Sect. 3, we analyze *Wang* et al.'s protocol. We present our practical attacks against *Safkhani* et al.'s protocol is Sect. 4. In Sect. 5, we show that *Sun* and *Zhong*'s protocol suffers from forward traceability attack. Then, in order to omit all weaknesses of the studied protocols, we propose improved versions of them which is reported in Sect. 6. We conclude the paper in Sect. 7.

2 Privacy Model and Related Attacks

2.1 Ouafi-Phan Privacy Model

In [28], *Ouafi* and *Phan* presented a formal model to evaluate the privacy of RFID authentications protocols. This model is an essential tool in the presented privacy analysis. In this model, the attacker $\mathcal{A}$ can eavesdrop all channels between target tags and readers and also he/she can perform active and passive attacks on them. In addition, the attacker $\mathcal{A}$ is allowed to run the following queries,

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

Send query (U, V, m, i): This query models the active attacks in RFID systems. In this query, the attacker $\mathcal{A}$ has permission to impersonate a reader U in the ith session, and forwards a message m to a 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.
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 database. In addition, the attacker $\mathcal{A}$ can set secret key to K'.
Test query (T_0, T_1, i): When this query is executed in the particular session i, after completing ith session**,** a random number bit $b \in \{0,1\}$ is generated by challenger and $T_b \in \{T_0, T_1\}$ is delivered to the attacker. Now, the attacker succeeds if he/she can guess the bit b correctly.

Partnership. A reader instance R_j and a tag instance T_i are partners if, and only if, both output *Accept* T_i and *Accept* R_i respectively, signifying the completion of the protocol session.

Freshness. A party instance is fresh at the end of execution if, and only if: i) it outputs *Accept* with or without a partner instance, ii) both the instance and its partner instance (if such a partner exists) have not received a *Corrupt* query.

Untraceability privacy (UPriv): Untraceability privacy could be defined by the game *G* that is played between an attacker $\mathcal{A}$ and a set of the tag and the 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:

- ***Learning phase:*** The attacker $\mathcal{A}$ has permission to send an *Execute*/*Send*/*Corrupt* query and interact with the reader R and T_0, T_1 that are chosen randomly.
- ***Challenge phase:*** The attacker $\mathcal{A}$ selects two tags T_0, 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 $\mathcal{A}$ receives a tag $T_b \in \{T_0, T_1\}$ using *Execute* and *Send* queries.
- ***Guess phase:*** Eventually, the attacker $\mathcal{A}$ finishes the game *G* and outputs a bit $b' \in \{0,1\}$ as guess of *b*.

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

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

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

2.2 Formal Privacy Attacks

In the privacy context, an RFID security scheme must be secure against three main privacy attacks including backward traceability, forward traceability and traceability. The main concept of these attacks can be expressed as follows.

Backward Traceability. In some application of RFID systems, it is necessary that an attacker would not be able to trace the location of a specific tag in the prior challenges; this property is defined as backward untraceability. More precisely, if an authentication protocol be vulnerable to backward traceability attack, an attacker can eavesdrop exchanged messages over communication channel of an specific tag and the reader and uses them to discover previous location of the target tag. This property can be provided via suitable updating of the secret values [11].

Forward Traceability. For any RFID authentication protocol, being secure against forward traceability attack is an exigent property which should be managed in the initial phases of designing. An RFID authentication protocol which provides forward untraceability is able to prevent tracing the location of a particular tag in the upcoming challenges. From the formal privacy model's point of view, if an attacker corrupts somehow secret values of a specific tag, he/she should not be able to track the location of the tag in future executions [11].

Traceability. Providing the end-user's privacy is one of the primary goals of each security protocol in different applications. Similarly, in an RFID authentication protocol, it is very important that an attacker would not be able to trace a specific tag if he/she has had access to the exchanged messages between the tag and a valid reader before last successful authentication. This property is known as untracability in formal privacy model. An RFID authentication protocol can provide untraceability property if tag's responses in two consecutive challenge are randomized and uncorrelated [11].

3 Analysis of Wang et al.'s Protocol

In this section, we cryptanalysis *Wang et al.*'s protocol [13]. It is shown that, although *Wang et al.* have tried to omit all weaknesses of *Pang et al.*'s protocol [17], still their improved protocol has some security and privacy weaknesses and

cannot provide security and privacy requirements of RFID end-users. To this aim, first we review *Wang et al.*'s protocol and then present our analyses on their protocol. The notations that are used in the paper are given in Table 1.

3.1 Wang et al.'s Protocol

Recently in [13], *Wang et al.* proposed an improved RFID authentication protocol which is under EPC C1 G2 standard. The structure of *Wang et al.*'s protocol is illustrated in Fig. 3. In their protocol, communication channel between the tag and the reader is insecure and can be eavesdropped by an attacker. The authentication procedure of the protocol is summarized in the rest of subsection.

Table 1. The Notations

Not.	Description	Not.	Description
EPC	Electronic Product Code	**PRNG**	Pseudo random number generator
$\mathbf{C_i}$	The database index stored in the tag to find the corresponding record of the tag in the reader.	$\mathbf{K_i}$	The authentication key stored in the tag to be used by reader to authenticate the tag at the $(i+1)th$ authentication phase.
$\mathbf{P_i}$	The tag's ith prescription recorded by the back-end database.	**X|a~b**	A fraction of string X includes bit b to bit a, where a > b.
$\mathbf{DID_T}$	The database index stored in the tag to find the corresponding record of the tag in the database.	$\mathbf{Asc_{RT}}$	The required proof to confirm that the current reader has the authority to access the tag stored in the tag only.
$\mathbf{Key_T}$	The key of the tag.	$\mathbf{Key_R}$	The key of the reader
$\mathbf{Key_s}$	The key of the back-end database.	$\mathbf{HP_i}$	The pseudonym value of prescription P_i.
$\mathbf{HC_i}$	The prescription's hash chain.	$\mathbf{ID_R}$	The identifier of the reader.
$\mathbf{ID_T}$	The identifier of the tag.	**ID**	The identifier of the tag.
$\mathbf{E_K}(.)$	A symmetric encryption function which uses K to encrypt the message.	$\mathbf{D_K}(.)$	A symmetric decryption function which uses K to decrypt the message.
LT(M)	represents the left half of the input message m.	**RT(M)**	represents the right half of the input message *m*.

The *Wang et al.*'s protocol consists of five steps which can be summarized as follows:

Step 0: Enrollment phase

a) In this phase, the secret value EPC_s and initial secret values such as K_0 and C_0 that are generated randomly in the manufacture, are shared between the tag and the reader. Also, the corresponding values of the mentioned parameters in the reader are set to these initial values ($K_{old} = K_{new} = K_0$ and $C_{old} = C_{new} = C_0$).

Step 1: The reader transmits a random number $N_1$to the tag.

Step 2: Response of the tag

a) The tag generates a random number N_2

b) Then, the tag computes and sends M_1, C_i, and M_2, to the reader as follows:

$$M_1 = N_2 \oplus PRNG(EPC_s \oplus K_i \oplus N_1),\ M_2 = PRNG(EPC_s \oplus N_2 \oplus C_i) \oplus K_i;$$

Step 3: The tag authentication

a) After receiving messages $\langle M_1, M_2, C_i \rangle$, firstly the reader matches C_{old} and C_{new} which has in its database with the received C_i and sets index i as "old" or "new".

b) After that, by using stored EPC_s and corresponding K_i of the legitimate tag, the reader calculates $N_2 = M_1 \oplus PRNG(EPC_s \oplus K_i \oplus N_1)$.

c) Then, the reader verifies that $M_2 \oplus K_i \stackrel{?}{=} PRNG(EPC_s \oplus N_2 \oplus C_i)$ to authenticate the tag. If the answer is "No", it aborts the rest of protocol.

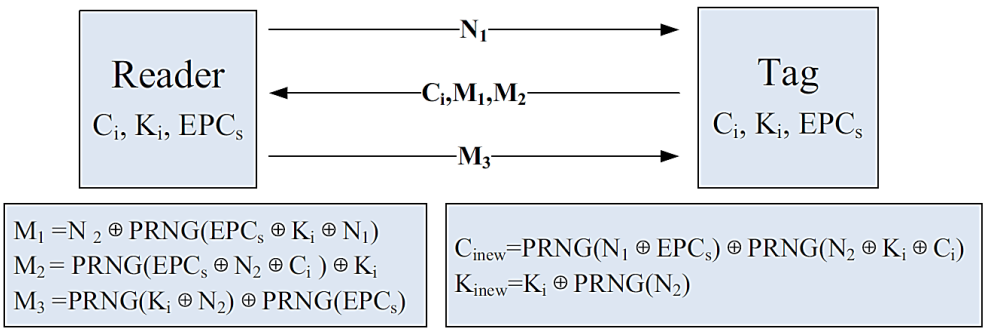

Fig. 3. Wang et al.'s protocol [13].

d) Then, $M_3 = PRNG(K_i \oplus N_2) \oplus PRNG(EPC_s)$ is calculated by reader and is sent to the tag and updates its secret values as follows,

$$C_{old} \leftarrow C_{new} \leftarrow PRNG(EPC_s \oplus N_1) \oplus PRNG(N_2 \oplus K_i \oplus C_i)$$
$$K_{old} \leftarrow K_{new} \leftarrow K_i \oplus PRNG(N_2)$$

Step 4: The Reader authentication

a) After receiving M_3 firstly, $M_3 \oplus PRNG(EPC_s) \stackrel{?}{=} PRNG(K_i \oplus N_2)$ is verified by the tag using his/her EPC_s, K_i and N_2. If the answer is "No", the tag aborts the rest of protocol, otherwise it authenticates the reader and updates its secret values as follows,

$$C_{i+1} \leftarrow PRNG(EPC_s \oplus N_1) \oplus PRNG(N_2 \oplus K_i \oplus C_i), \qquad K_{i+1} \leftarrow K_i \oplus PRNG(N_2)$$

3.2 Secret Parameter Recovery

This subsection aims to show that an adversary can recover all secret parameters (EPC_s, K_i) of *Wang et al.*'s protocol. This attack consists of two phases as follows:

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

$$M_{1,i} = N_{2,i} \oplus PRNG(EPC_s \oplus K_i \oplus N_1), \quad M_{2,i} = PRNG\big(EPC_s \oplus N_{2,i} \oplus C_i\big) \oplus K_i$$

Attack phase: In the next session, the attacker starts a new session with the target tag and receives $M_{1,i+1} = N_{2,i+1} \oplus PRNG(EPC_s \oplus K_{i+1} \oplus N_1)$ and $M_{2,i+1} = PRNG\big(EPC_s \oplus N_{2,i+1} \oplus C_{i+1}\big) \oplus K_{i+1}$ by sending N_1. Then, the attacker aborts the rest of protocol. After that , the attacker uses $M_{1,i+1}$, $M_{2,i+1}$ and the obtained data in the learning phase and performs the following steps,

1) The attacker calculates $M_{1,i} \oplus M_{1,i+1}$ and $M_{2,i} \oplus M_{2,i+1}$ as follows

$$M_{1,i} \oplus M_{1,i+1} = N_{2,i} \oplus PRNG(EPC_s \oplus K_i \oplus N_1) \oplus N_{2,i+1} \oplus PRNG(EPC_s \oplus K_{i+1} \oplus N_1) = N_{2,i} \oplus N_{2,i+1} \quad (1)$$

$$M_{2,i} \oplus M_{2,i+1} = PRNG\big(EPC_s \oplus N_{2,i} \oplus C_i\big) \oplus K_i \oplus PRNG\big(EPC_s \oplus N_{2,i+1} \oplus C_{i+1}\big) \oplus K_{i+1}. \quad (2)$$

Let $\alpha = EPC_s \oplus N_{2,i} \oplus C_i$ and $\beta = EPC_s \oplus N_{2,i+1} \oplus C_{i+1}$. Equation (2) can then be rewritten as follows,

$$M_{2,i} \oplus M_{2,i+1} = PRNG(\alpha) \oplus PRNG(\beta) \quad (3)$$

It can be observed that,

$$\begin{aligned} \alpha \oplus \beta &= EPC_s \oplus N_{2,i} \oplus C_i \oplus EPC_s \oplus N_{2,i+1} \oplus C_{i+1} \\ &= N_{2,i} \oplus N_{2,i+1} \\ &= M_{1,i} \oplus M_{1,i+1} \end{aligned}$$

As a result, we have $\beta = M_{1,i} \oplus M_{1,i+1} \oplus \alpha$ and (2) can be rewritten as follows,

$$M_{2,i} \oplus M_{2,i+1} = PRNG(\alpha) \oplus PRNG\big(M_{1,i} \oplus M_{1,i+1} \oplus \alpha\big)$$

Since α is a 16-bit string, the correct value can be found by trying all 2^{16} possible values.

2) Now, using $M_{2,i}$ and the obtained α, the value of K_i can be calculated as follows,

$$K_i = PRNG(\alpha) \oplus M_{2,i}$$

$$= PRNG\left(EPC_s \oplus N_{2,i} \oplus C_i\right) \oplus M_{2,i}$$

3) Now using K_i, $M_{1,i}$, N_1 and C_i that are obtained in the previous steps, we get

$$M_{1,i} \oplus \alpha \oplus C_i = N_{2,i} \oplus PRNG(EPC_s \oplus K_i \oplus N_1) \oplus EPC_s \oplus N_{2,i} \oplus C_i \oplus C_i$$
$$= PRNG(EPC_s \oplus K_i \oplus N_1) \oplus EPC_s$$

The only unknown variable, EPC_s, can be found by comprehensive search all 2^{16} possible values.
It can be seen that in order to perform this attack, the adversary needs to eavesdrop one session of the protocol and 2×2^{16} *PRNG* computations. It is worth to mention that after obtaining all secret values of the tag, the adversary can perform various attacks such as, traceability, tag impersonation, reader impersonation, and DoS attacks with the success probability of "1". Furthermore, *Wang et al.*'s protocol has some problems that in the rest of paper some of the possible attacks are provided.

3.3 Traceability Attack

One of the major problems in *Wang et al.*'s protocol is the fact that the tag updates its parameter C_{new}, after a successful authentication. Here, we show that an adversary can use this fact as a weakness and trace a target tag as follows,

Learning phase: In round (i), the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$ to the tag, and obtains $C_i^{T_0}$ after that the attacker $\mathcal{A}$ sends a $Send\ query(R, T_0, i)$, and blocks protocol. As a result the tag does not update secret values.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(T_0, T_1, i+1)$. According to the randomly chosen bit $b \in \{0,1\}$, 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_1 message, and obtains $C_{i+1}^{T_b}$.
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.

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

Therefore, $Adv_A^{upriv}(k) = \left|pr(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \varepsilon$.

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

3.4 Forward Traceability Attack

In this part, it is shown that *Wang et al.*'s protocol also does not provide forward privacy and an adversary can perform forward traceability attack as follows:
Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $\left(EPC_{s,i}^{T_0}, K_i^{T_0}\right)$ from tag T_0. After that, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_0, i)$ and obtains $\left(M_{1,i}^{T_0}, C_i^{T_0}, N_i^{T_0}\right)$. Then he/she computes $\psi = PRNG\left(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus N_1^{T_0}\right)$ and $\zeta = K_i^{T_0} \oplus PRNG(\psi \oplus M_{1,i}^{T_0})$.
Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test and sends a $Test\ query(T_0, T_1, i+1)$. 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, in round $(i+1)$, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+1)$, by sending $N_i^{T_0}$, and obtains $M_{1,i+1}^{T_b}$ and $M_{2,i+1}^{T_b}$.

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 guess b', first the attacker $\mathcal{A}$ computes $\eta = M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus \zeta \oplus N_1^{T_0})$ and $\chi = PRNG(EPC_{s,i}^{T_0} \oplus \eta \oplus C_1^{T_b}) \oplus \zeta$. Then, the attacker $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as a guess of bit b using the following rule:

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

As a result, $Adv_A^{upriv}(k) = \left|pr(b' = b) - \frac{1}{2}\right| = \left|1 - \frac{1}{2}\right| = \frac{1}{2} \gg \varepsilon$.

Proof: Since the value of EPC_s is fixed in all rounds, thus $EPC_{s,i}^{T_0} = EPC_{s,i+1}^{T_0}$. Using this fact, and assuming $T_b = T_0$, we have

$$\chi = PRNG(EPC_{s,i}^{T_0} \oplus \eta \oplus C_1^{T_b}) \oplus \zeta \tag{4}$$

By substituting $\eta = M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus \zeta \oplus N_1^{T_0})$ and $\zeta = K_i^{T_0} \oplus PRNG(\psi \oplus M_{1,i}^{T_0})$, we then get

$$\chi = PRNG(EPC_{s,i}^{T_0} \oplus M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus PRNG(\psi \oplus M_{1,i}^{T_0}) \oplus N_1^{T_0}) \oplus C_1^{T_b}) \oplus K_i^{T_0} \oplus PRNG(\psi \oplus M_{1,i}^{T_0}). \quad (5)$$

Using the fact that $\psi = PRNG\left(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus N_1^{T_0}\right)$, we can write

$$\chi = PRNG(EPC_{s,i}^{T_0} \oplus M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus PRNG(PRNG(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus N_1^{T_0}) \oplus M_{1,i}^{T_0}) \oplus N_1^{T_0}) \oplus C_1^{T_b}) \oplus K_i^{T_0} \oplus PRNG(PRNG\left(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus N_1^{T_0}\right) \oplus M_{1,i}^{T_0}). \quad (6)$$

According to the protocol, $N_{2,i}^{T_0} = PRNG\left(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus N_1^{T_0}\right) \oplus M_{1,i}^{T_0}$, thus,

$$\chi = PRNG(EPC_{s,i}^{T_0} \oplus M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus K_i^{T_0} \oplus PRNG(N_{2,i}^{T_0}) \oplus N_1^{T_0}) \oplus C_1^{T_b}) \oplus K_i^{T_0} \oplus PRNG(N_{2,i}^{T_0}) \quad (7)$$

By substituting the updated value of $K_{i+1}^{T_0} = K_i^{T_0} \oplus PRNG(N_{2,i}^{T_0})$, we have

$$\chi = PRNG\left(EPC_{s,i}^{T_0} \oplus M_{i+1}^{T_b} \oplus PRNG\left(EPC_{s,i}^{T_0} \oplus K_{i+1}^{T_0} \oplus N_1^{T_0}\right) \oplus C_1^{T_b}\right) \oplus K_{i+1}^{T_0}. \quad (8)$$

Finally, with substituting the values of $T_b = T_0$ and $N_{2,i+1}^{T_b} = M_{i+1}^{T_b} \oplus PRNG(EPC_{s,i}^{T_0} \oplus K_{i+1}^{T_b} \oplus N_1^{T_0})$ can be rewritten as follows,

$$\chi = PRNG(EPC_{s,i}^{T_b} \oplus N_{2,i+1}^{T_b} \oplus C_1^{T_b}) \oplus K_{i+1}^{T_b} = M_{1,i+1}^{T_b}. \quad (9)$$

4 Analysis of Safkhani et al.'s Protocol

Providing secure and confidential communication for end-users is the most prominent goal of each RFID authentication protocol. In this section, we investigate the privacy of *Safkhani et al.*'s protocol [21] and point out that their protocol is not resistant against traceability and forward traceability attacks.

4.1 Safkhani et al.'s Protocol

In [21], *Safkhani et al.* proposed an improved RFID authentication protocol which uses a symmetric cryptosystem to protect RFID users. Each run of their protocol consists of five Steps which is shown in Fig. 4 and are given in the rest of subsection with more details.

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

Step 2. Upon receiving N_R, the tag generates a random number N_T and calculates the following messages and sends the triple (DID_T, N_T, V_T) to the reader.

$$x_T = (ACS_{RT} \oplus N_R) \parallel (HP_{i-1} \oplus N_T), \quad y_T = Key_T \oplus DID_T, \quad V_T = E_{y_T}(x_T).$$

Step 3. The reader computes x_R, C_r, y_R and V_R as follows and transmits ($DID_T, ID_R, N_T, N_R, V_T, V_R$ and C_R) to the back-end server.

$$x_R = (M) \parallel (N_R \oplus N_T), \quad C_r = C_r + 1, \quad y_R = Key_R \oplus C_r, \quad V_R = E_{y_R}(x_R).$$

Step 4. Now, using the received messages from the reader, the back-end server performs the following operations:

1) Aaccording to DID_T, it retrieves tag information including ACS_{RT} and HP_{i-1}. Then it computes messages $x_T = (Acs_{RT} \oplus N_R) \parallel (HP_{i-1} \oplus N_T)$, $y_T = Key_T \oplus DID_T$, $x_R = (M) \parallel (N_R \oplus N_T)$,$y_R = Key_R \oplus C_r$.
2) The server verifies $x_T \stackrel{?}{=} (Acs_{RT} \oplus N_R) \parallel (HP_{i-1} \oplus N_T)$ and $D_{y_R}(V_R)\,|_{(l-1)-0} \stackrel{?}{=} N_R \oplus N_T$ and $C'_r > C_r$ and follows the rest of authentication procedure. Then it updates its secret values as follows,

$$C'_r = C_r, \quad P_i = D_{y_R}(V_R)\,|_{(2l-1)-l}, \quad DID_T = h(ID_T \parallel N_T),$$

$$HP_i = h(P_i), \quad HC_i = h(P_{i-1}, P_i).$$

where $h(.)$ is a one-way hash function.

3) Finally, the back-end server uses the updating secret values and received messages from the reader, and it computes following messages and send V_s to the reader.

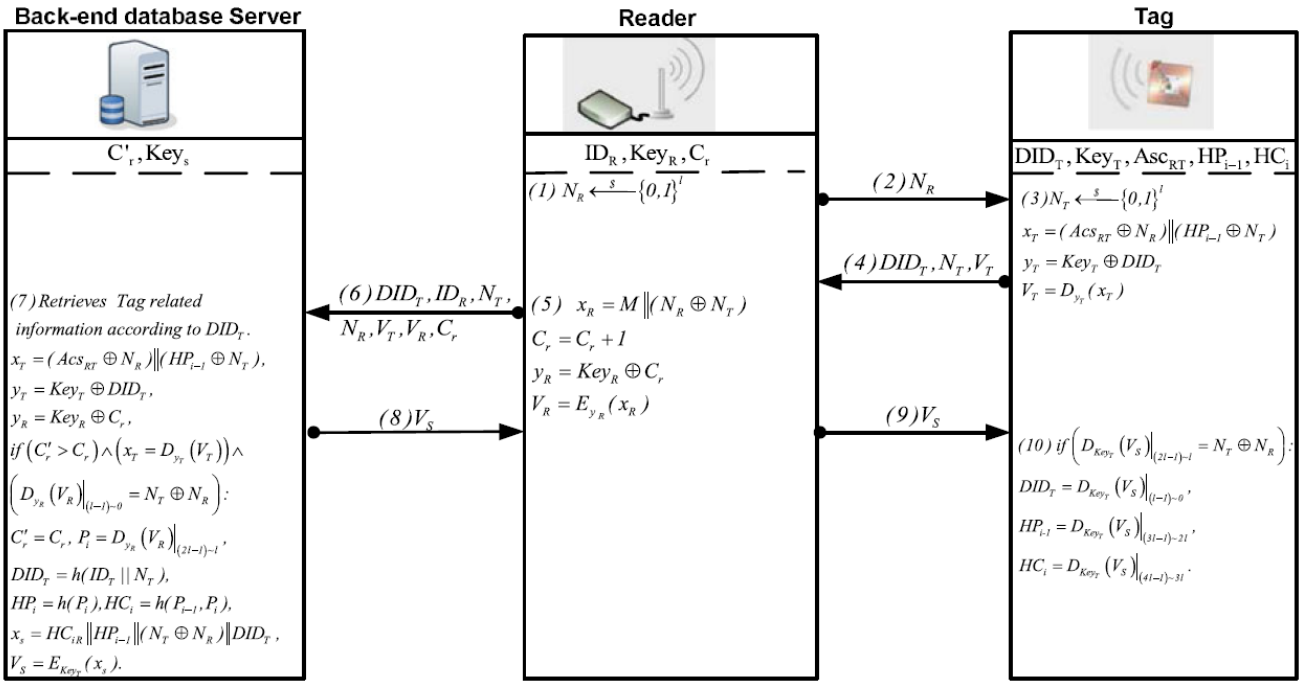

Fig. 4. Safkhani et al.'s protocol [21].

$$x_s = HC_i \parallel HP_i \parallel N_R \oplus N_T \parallel DID_T, \qquad V_s = E_{Key_T}(x_s).$$

Step 5. The reader transmits message V_s to the tag. Upon receiving the message, the tag verifies $D_{Key_T}(V_s)\,|_{(2l-1)-l} \stackrel{?}{=} N_R \oplus N_T$. If the answer is "Yes", it updates the secret values as follows,

$$DID_T = D_{Key_T}(V_s)\,|_{(l-1)-0}, \quad HP_{i-1} = D_{Key_T}(V_s)\,|_{(3l-1)-2l}, \quad HC_i = D_{Key_T}(V_s)\,|_{(4l-1)-3l}.$$

4.2 Traceability Attack

Providing an untraceable communication for end-users is one of primary goals for each RFID authentication protocol. This subsection aims to show that *Safkhani et al.*'s protocol does not ensure untraceability and an attacker can trace a target tag as follows,

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

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for test, and sends a $Test\ query(T_0, T_1, i+1)$. 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 then the attacker obtains $DID_{T,i+1}^{T_b}$.

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.

$$b' = \begin{cases} 0 & if\ DID_{T,i+1}^{T_b} = DID_{T,i}^{T_0} \\ 1 & otherwise \end{cases}$$

As a result,

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

Proof: According to the structure of *Safkhani et al.*'s protocol, we can see which in the *Learning phase*, the tag T_0 does not update its secret values and uses the same secret value $DID_{T,i}^{T_0}$ in the both *Learning* and *Challenge phases* (i.e., rounds i and $i+1$).

4.3 Forward Traceability Attack

We show that *Safkhani et al.*'s protocol also does not assure the forward untraceability. According to the structure of *Safkhani et al.*'s protocol, it can be seen that the ID_T is fixed in all rounds. Using this fact, an attacker can trace a target tag as follows,

Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $(DID_{T,i}^{T_0}, Key_i^{T_0}, ID_{T,i}^{T_0}, N_{T,i}^{T_0})$ from tag T_0. It also sends an $Execute\ query(R, T_0, i)$ and obtains $N_{T,i}^{T_0}$.

Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(T_0, T_1, i)$. 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 $(i+1)$, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+1)$ by sending $N_{R,i}$ and obtains $DID_{T,i+1}^{T_b}$. Now the attacker can compute $DID_{T,i+1}$ at the session $i+1$ by computing $h(ID_T \parallel N_{T,i})$.

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 $\zeta = h(ID_{T,i}^{T_0} \parallel N_{T,i}^{T_0})$. Then, the attacker $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as a guess of bit b using the following rule.

$$b' = \begin{cases} 0 & if\ \ DID_{T,i+1}^{T_b} = \zeta \\ 1 & otherwise \end{cases}$$

As a result,

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

Proof: Since the value of ID_T is fixed in all rounds, thus $ID_{T,i}^{T_0} = ID_{T,i+1}^{T_0}$. Using this fact, the following equations can be written.

$$\begin{aligned} If\ \ T_b = T_0: \quad DID_{T,i+1}^{T_b} &= h(ID_{T,i+1}^{T_b} \parallel N_{T,i}^{T_b}) \\ &= h(ID_{T,i}^{T_0} \parallel N_{T,i}^{T_0}) = \zeta \end{aligned} \tag{10}$$

5 Analysis of Sun-Zhong Protocol

In 2012, *Sun* and *Zhong* [23] analyzed a hash-based RFID authentication protocol which was proposed by *Ha et al.* [22], and proposed a strengthened version of *Ha et al.*'s protocol. The *Sun-Zhong* protocol is hash based and protects the exchanged messages between the tag and the reader by hash functions. Fig. 5 shows the structure of the *Sun-Zhong* protocol. In this section, first we review the *Sun-Zhong* protocol and then present a backward traceability attack against it.

5.1 Sun-Zhong Protocol

Step 1. The reader generates a random number r_R and sends it to the tag with Query.
Step 2. Upon receiving messages from the reader, the tag generates r_T and computes $Q = H_l(ID \parallel r_T \parallel r_R)$ and sends r_T and $L(Q)$ to the reader.
Step 3. In order to authenticate the tag, the reader calculates $Q' = LT(H_l(H^i(ID) \parallel r_{T,1} \parallel r_R))$ satisfying $0 \le i \le t$, then verifies $LT(Q') \stackrel{?}{=} LT(Q)$. After that, the reader calculates $RT(Q')$ and transmits it to the tag. Finally he/she updates $ID = H^i(ID)$ for next run.
Step 4. Upon receiving the message $RT(Q')$ from the reader, the tag verifies $RT(Q) \stackrel{?}{=} RT(Q')$ to authenticate the reader; if the tag does not authenticate the reader successfully, it terminates the session.

5.2 Forward Traceability Attack

In [23], *Sun* and *Zhong* claimed that their protocol provides strong privacy for RFID users. However, in this subsection we aim to show that it is not safe against forward traceability attack. This attack is performed as follows:
Learning phase: In the ith round, the attacker $\mathcal{A}$ sends a $Corrupt\ query(T_0, K')$ and obtains $\left(ID_i^{T_0}\right)$ from tag T_0. Now the attacker can compute $ID_{i+2}^{T_0}$ at the session $i+2$ by applying the hash function two times on $ID_i^{T_0}$.
Challenge phase: The attacker $\mathcal{A}$ selects two fresh tags T_0 and T_1 for the test, and sends a $Test\ query(T_0, T_1, i+2)$. According to the randomly chosen bit $b \in \{0,1\}$, the attacker is given a tag $T_b \in \{T_0, T_1\}$. After that, in round $(i+2)$, the attacker $\mathcal{A}$ sends an $Execute\ query(R, T_b, i+2)$ by sending $r_{R,i}^{T_0}$ (i.e., the same value as for session i) and obtains $\left(L\left(Q_{i+2}^{T_b}\right), r_{T,i+2}^{T_b}\right)$.
Guess phase: The attacker $\mathcal{A}$ stops the game G, and outputs a bit $b' \in \{0,1\}$ as a guess of bit b. In order to guess b', firstly the attacker $\mathcal{A}$ computes $\alpha = ID_{i+2}^{T_0} = H\left(H\left(ID_i^{T_0}\right)\right)$, $\beta = H\left(\alpha \oplus r_{T,i+2}^{T_b} \oplus r_{R,i}^{T_0}\right)$ and $\gamma = L(\beta)$. Then, he/she outputs a bit $b' \in \{0,1\}$ as a guess of bit b using the following rule:

$$b' = \begin{cases} 0 & if\ \ L\left(Q_{i+2}^{T_b}\right) = \gamma \\ 1 & otherwise \end{cases}.$$

Reader (ID)		**Tag** (ID)
	Query, $r_R \rightarrow$	$Q = H_l(ID \parallel r_T \parallel r_R)$ $ID = H(ID)$
If $\{LT(H_l(H^i(ID) \parallel r_T \parallel r_R)) = LT(Q)\}$ and $0 \le i \le t$ $\{ID = H^i(ID), Q' = H_l(H^i(ID) \parallel r_T \parallel r_R),$ $Successfully\ complet\ the\ session\}$ *Else* $\{Unsuccessfully\ terminate\ the\ session\}$	$\leftarrow (L(Q), r_T)$ $R(Q') \rightarrow$	If $RT(Q) \stackrel{?}{=} RT(Q')$ $\{Successfully\ complet\ the\ session\}$ *Else* $\{Unsuccessfully\ terminate\ the\ session\}$

Fig. 5. The *Sun-Zhong* protocol [23].

As a result,

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

Notice that the attacker can obtain $ID_{i+n}^{T_0}$ for $n \geq 1$ using $ID_i^{T_0}$.

Proof: Since the value of ID is fixed in all rounds, thus $ID_i^{T_0} = ID_{i+2}^{T_0}$. Using this fact, assuming $T_b = T_0$, the following equations can be written:

$$\begin{aligned} \gamma = L(\beta) &= L\left(H\left(\alpha \oplus r_{T,i+2}^{T_b} \oplus r_{R,i}^{T_0}\right)\right) \\ &= L\left(H\left(H\left(H\left(ID_i^{T_0}\right)\right) \oplus r_{T,i+2}^{T_b} \oplus r_{R,i}^{T_0}\right)\right). \end{aligned} \tag{11}$$

Since $ID_{i+2}^{T_0} = H\left(H\left(ID_i^{T_0}\right)\right)$, equation (11) can be written as follows:

$$\gamma = L\left(H\left(ID_{i+2}^{T_0} \oplus r_{T,i+2}^{T_b} \oplus r_{R,i}^{T_0}\right)\right)$$

Eventually, if $T_b = T_0$, we can conclude that $ID_{i+2}^{T_0} = ID_{i+2}^{T_b}$. So we have,

$$\begin{aligned} \gamma &= L\left(H\left(ID_{i+2}^{T_b} \oplus r_{T,i+2}^{T_b} \oplus r_{R,i}^{T_0}\right)\right) \\ &= L\left(Q_{i+2}^{T_b}\right). \end{aligned} \tag{12}$$

6 Improved Versions of the Analyzed Protocols

In sections 3, 4 and 5 it is shown that *Wang et al.*'s, *Safkhani et al.*'s and *Sun-Zhong* protocols have some problems which make them vulnerable to various traceability attacks. In this section, in order to overcome all the reported weaknesses we apply some modifications in the analyzed protocols and present an improved version of each one.

6.1 Improved Version of Wang et al.'s Protocol

In this subsection, in order to eliminate all the mentioned weaknesses of *Wang et al.*'s protocol which was presented in Section 3, we apply some modification on its structure and propose a modified version. There are two main problems in the structure of *Wang et al.*'s protocol. First one is dependency between tag's responses including M_1 and M_2 which made the protocol vulnerable to secret parameters reveal and information leakage of the tag. The second one is updating procedure of secret keys in the tag and the reader which makes privacy concerns.

In order to remove the mentioned weaknesses and prevent the presented attacks, we apply some changes in the structure of *Wang et al.*'s protocol. First, we change computing methods of M_1 and M_2 as $M_1^{new} = N_2 \oplus PRNG(EPC_s \oplus N_3 \oplus N_1)$, $M_2^{new} = PRNG(N_2 \oplus C_i) \oplus K_i$, where N_3 is a new random number that is generated in the tag. Another change which increases the privacy of the *Wang et al.*'s protocol is to the update C_i and K_i as follows,

$$C_{i+1} \leftarrow PRNG(C_i \oplus N_2),\ K_{i+1} \leftarrow PRNG(K_i \oplus N_2 \oplus N_3).$$

In addition, we propose a modification in the tag response C_i. We define $C_{i,sent} = N_3 \oplus C_i$ where N_3 is a new random number generated by the tag. After applying all the proposed modifications, final structure of improved protocol is shown in Fig. 6. Now we analyze that how the proposed modifications overcome all the discovered drawbacks and make the protocol resistant against various security and privacy attacks.

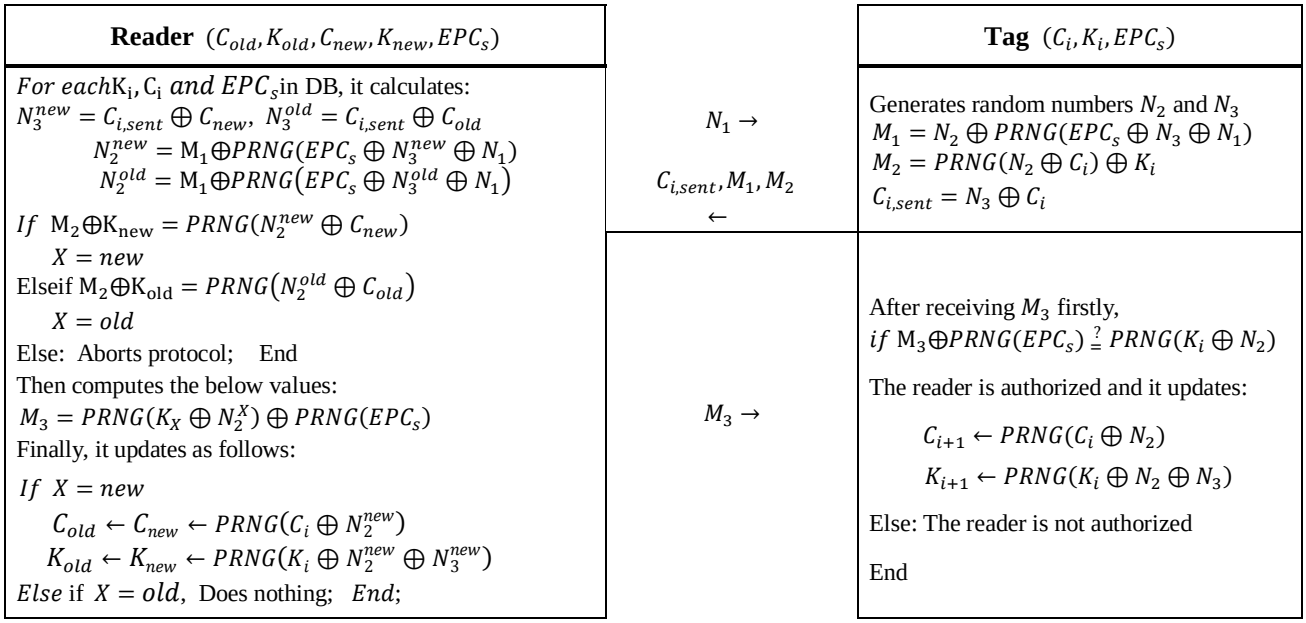

Fig. 6. Improved version of *Wang et al.*'s protocol.

- **Secret parameter reveal**

In [13], *Wang et al.* showed that due to some weaknesses in the tag responses CN_2, M_1, M_2 and updating of the secret key K_i in the tag, *Pang et al.*'s [17] protocol is vulnerable to secret parameter reveal attack and an attacker can obtain secret keys with $O(2^{16})$ attack complexity. In Subsection 3.2, we showed that there is another weakness in the tag responses of the *Wang et al.*'s protocol which is the strengthened version of *Pang et al.*'s protocol. This weakness arises from dependency between consecutive tag responses including M_1^i, M_2^i, M_1^{i+1} and M_2^{i+1}. In the improved version of *Wang et al.*'s protocol, we propose some modifications in the messages M_1 and M_2 which eliminate the mentioned weaknesses and prevent both the presented secret parameters reveal attacks presented in [13] and Subsection 3.2. In fact, with the proposed modifications, not only tag responses in two consecutive runs become independent each other, but also the random values in the tag responses are increased and consequently the complexity of attack increases significantly.

- **Traceability**

In Subsection 3.3, we showed that how an attacker can use the weakness on the structure of C_i and its updating procedure and performs traceability attack. In the modified protocol, we remove these weaknesses by two changes in the updating of C_i and in the structure of transmitted C_i in the tag responses. We use the random number N_2 in the updating of C_i as $C_{i+1} \leftarrow PRNG(C_i \oplus N_2)$. With this change, after each successful authentication, the tag updates its secret value with a new random number which prevents the attacker from predicting the next C_i. Moreover, we modify the value of C_i in the tag responses; in other words, we XOR a new random number N_3 with the transmitted C_i in the tag responses. Note that N_3 is a random number generated by the tag in each new challenge. With the second change, if an attacker blocks a phase of protocol before successful authentication and starts a new challenge with the tag, the tag will response a new C_i which overcomes the existing weaknesses and make the improved protocol more secure than before. As a result, the improved protocol prevents traceability attacks and an attacker cannot trace the current location of a specific tag.

- **Forward traceability**

According to the presented forward traceability attack in Subsection 3.4, we observed that in the *Wang et al.*'s protocol, there are some drawbacks in the updating of secret keys K_i and C_i, and the structure of tag response M_1 which makes the attacker able to trace the location of a specific tag in the next runs. In the *Wang et al.*'s protocol, if an attacker corrupts the secret keys and uses the eavesdropped messages, he/she can compute $\psi = PRNG(EPC_{S,i} \oplus K_i \oplus N_1)$ and obtain N_2. Then using the obtained N_2, he/she can calculate K_{i+1} which is the secret key of a specific tag in the next run. In the improved protocol, in order to overcome this weakness we change the updating procedures of the mentioned secret keys and the structure of the message M_1. We use the random number N_3 in the updating of the K_i which increases the privacy of protocol. With these modifications in the updating procedure, if an attacker corrupts the secret keys, he/she will not be able to calculate random numbers N_2 and N_3 using the message M_1. Consequently, the attacker cannot perform forward traceability attack and trace the location of a specific tag in the next runs.

6.2 Improved Version of Safkhani et al.'s Protocol

Similar to the *Wang et al.*'s protocol, *Safkhani et al.*'s protocol suffers from two main privacy problems making it vulnerable to traceability and forward traceability attacks. It can be shown, that with two changes in the tag's responses and updating procedure of secret keys, both the mentioned weaknesses will be omitted. In the tag's responses we define a new variable $DID_i^{sent} = DID_i \oplus N_3$ which in each new run of the protocol the tag transmits to the reader. The variable DID_i is a dynamic identifier of the tag which is updated after each successful run of the protocol and N_3 is a new random number which is generated in the tag. Moreover, in order to prevent forward traceability attack, we change the updating of $DID_T = h(ID_T \| N_T)$ as $DID_T^{new} = h(ID_T \| (N_T \oplus N_3))$. All the identification and authentication steps of the improved protocol are similar to *Safkhani et al.*'s protocol. That is, only we change the values of DID_i and updating of DID_T in Step 2 and Step 5, respectively. The authentication steps of the improved protocol, shown in Fig. 7, can be expressed as follows:

Step 1. This step is same as *Safkhani et al.*'s protocol.

Step 2. Upon receiving N_R, the tag generates two random numbers N_T and N_3 and calculates x_T, y_T, and V_T similar to *Safkhani et al.*'s protocol and sends them with a new variable DID_T^{sent} to the reader, where $DID_T^{sent} = DID_T \oplus N_3$.

Step 3. Same as *Safkhani et al.*'s protocol.

Step 4. Now, using the received messages from the reader, the back-end server performs the following operations:

1) It computes messages $x_R = (M) \| (N_R \oplus N_T)$, $y_R = Key_R \oplus C_r$, $x_{T,q} = \left(ACS_{RT,q} \oplus N_R\right) \| \left(HP_{i-1,q} \oplus N_T\right)$, $y_{T,q} = Key_T \oplus DID_{T,q}$, for $q = new\ and\ old$.
2) It verifies $x_{T,q} \stackrel{?}{=} D_{y_{T,q}}(V_T)$, $D_{y_R}(V_R)\ |_{(l-1)-0} \stackrel{?}{=} N_R \oplus N_T$ and $C'_r > C_r$, and follows the rest of authentication procedure and determines q as *old* or *new*. After that it updates secret values C'_r, P_i, HP_i, and HC_i similar to *Safkhani*

Server	**Reader**	**Tag**
$(Key_R, Key_T, DID_{old}, ACS_{RT,old}, HP_{i-1,\text{old}}, HC_{i,old}, DID_{T,old}, ACS_{RT,new}, HP_{i-1,\text{new}}, HC_{i,new}, C'_r)$	(ID_R, Key_R, C_r)	$(DID_T, Key_T, ACS_{RT}, HP_{i-1}, HC_i)$
$For\ each\ DID_{old}$ and DID_{new} in DB, it computes: $N_3^{new} = DID_T^{sent} \oplus DID_T^{new}$ $N_3^{old} = DID_T^{sent} \oplus DID_T^{old}$ $Then\ the\ server\ to\ authenticate\ the\ reader\ acts\ as\ followes,$ $x_R = (M) \| (N_R \oplus N_T)$ $y_R = Key_R \oplus C_r$ $verifies: x_R \stackrel{?}{=} D_{y_R}(V_R)$ $Then, the\ server\ retrieves\ tag\ related$ $For\ q = old$ and new $x_{T,q} = \left(ACS_{RT,q} \oplus N_R\right) \| \left(HP_{i-1,q} \oplus N_T\right)$ $y_{T,q} = Key_T \oplus DID_{T,q}$ $It\ verifies: x_{T,q} \stackrel{?}{=} D_{y_{T,q}}(V_T)$ $D_{y_R}(V_R)\ |_{(l-1)-0} \stackrel{?}{=} N_R \oplus N_T$ $C'_r > C_r$ $The\ server\ determines\ q = old\ or\ new,$ $and\ it\ authenticatesthes\ reader\ and\ the\ tag$ Then the server updates as follows: If $q = new$ $C'_r = C_r$ $P_{old} \leftarrow P_{new} \leftarrow D_{y_R}(V_R)\ |_{(2l-1)-l}$ $DID_{old} \leftarrow DID_{new} \leftarrow h(ID_T \| N_T \oplus N_3^q)$ $HP_{old} \leftarrow HP_{new} \leftarrow h(P_i)$ $HC_{old} \leftarrow HC_{new} \leftarrow h(P_{i-1}, P_i)$, End; $Finally\ it\ computes\ following\ messages\ and\ sends\ to\ the\ reader$: $x_s = HC_{new} \| HP_{new} \| N_R \oplus N_T \| DID_{new}$ $V_s = E_{Key_T}(x_s)$	$N_R \stackrel{(1)}{\rightarrow}$ $\stackrel{(2)}{\leftarrow} (DID_T^{sent}, N_T, V_T)$ $x_R = (M) \| (N_R \oplus N_T)$ $C_r = C_r + 1$ $y_R = Key_R \oplus C_r$ $V_R = E_{y_R}(x_R)$ $\stackrel{(3)}{\leftarrow} (DID_T^{sent}, N_T, V_T, ID_R, N_R, V_R, , C_r)$ $V_s \stackrel{(4)}{\rightarrow}$ $V_s \stackrel{(5)}{\rightarrow}$	Generates random numbers N_T and N_3 $x_T = (ACS_{RT} \oplus N_R) \| (HP_{i-1} \oplus N_T)$ $y_T = Key_T \oplus DID_T$ $V_T = E_{y_T}(x_T)$ $DID_T^{sent} = DID_{T,q} \oplus N_3$ $The\ tag\ verifies$: $D_{Key_T}(V_s)\ |_{(2l-1)-l} \stackrel{?}{=} N_R \oplus N_T$ $Fainally\ it\ updates\ as\ followes$: $DID_T = D_{Key_T}(V_s)\ |_{(l-1)-0}$ $HP_{i-1} = D_{Key_T}(V_s)\ |_{(3l-1)-2l}$ $HC_i = D_{Key_T}(V_s)\ |_{(4l-1)-3l}$

Fig. 7. The improved version of the Safkhani et al.'s protocol.

et al.'s protocol, but it updates DID_T as $DID_T = h(ID_T \parallel N_T \oplus DID_{T,q} \oplus DID_T^{sent})$.

3) Same as before.

Step 5. Same as *Safkhani et al.*'s protocol.

In the rest of subsection, we discuss how the proposed changes overcome both the presented traceability attacks.

- ***Traceability Attack:***

In Section 3, we observed that due to a weakness in the tag response DID_T, more precisely DID_T remains fix if an adversary terminates Step 2 of the protocol and starts a new session with the tag, *Safkhani et al.*'s protocol suffers from traceability attack. Now, in order to overcome this attack, a new variable $DID_T^{sent} = DID_T \oplus N_3$ is added to the tag's responses which will be updated in each new run of the protocol. The variable DID_T is a dynamic identifier of the tag which is updated after each successful run of the protocol and N_3 is a new random number which is generated in the tag. It can be seen that, the new applied change omits the mentioned weakness of *Safhani et al.*'s protocol which solves its privacy problem.

- ***Forward Traceability Attack:***

The next problem in the *Safkhani et al.*'s protocol was updating procedure of secret keys in the tag and the back-end server which made the protocol vulnerable to forward traceability attack. Our analysis showed that, if we modify the structure of updating procedure of DID_T as $DID_T^{new} = h(ID_T \parallel (N_T \oplus N_3))$, the mentioned concern will be omitted; where N_3 is a new random number which has generated by the tag.

6.3 Improved Version of Sun-Zhong Protocol

In the Section 5, it is shown that in the *Sun-Zhong* protocol an adversary can eavesdrop the random number r_T exchanged between the tag and the reader and uses it for forward traceability attack. To overcome this problem, we define a new variable K_i which is shared between the tag and the reader. With this modification, if an attacker eavesdrops the exchanged messages between the tag and the reader it will not be able to obtain r_T to perform attacks and achieve its wicked goals. Note that the back-end server stores the old and the new values of K_i to prevent DoS attack. Moreover, by updating the value of K_i in the tag and the back-end server, the protocol prevents traceability attack. The structure and the authentication phases of the improved version of *Sun-Zhong* protocol is shown in Fig. 8.

Server/Reader		**Tag**
(K_{old}, K_{new}, ID)		(ID, K_i)
For each K_i and ID in DB, it computes: $r_{T,1} = K_{i,T} \oplus K_{new}$, $r_{T,2} = K_{i,T} \oplus K_{old}$ *For* $0 \le j \le t$ *If* $\{LT(H_l(H^j(ID) \parallel r_{T,1} \parallel r_R)) = LT(Q)\} \Rightarrow X = new$ *Elseif* $\{LT(H_l(H^j(ID) \parallel r_{T,2} \parallel r_R)) = LT(Q)\} \Rightarrow X = old$ *Else* the tag is not authorized; *End*; Then it computes the value, $Q' = H_l(H^j(ID) \parallel r_{T,X} \parallel r_R)$ Then it updates its secret values as follows, *If* $X = new$: $ID \leftarrow H^j(ID)$, $K_{old} \leftarrow K_{new} \leftarrow H(r_T \oplus K_i)$ *Else* $X = old$: $ID \leftarrow H^j(ID)$. *End*;	$r_R \rightarrow$ $\leftarrow (LT(Q), K_{i,T})$ $Q' \rightarrow$	Generates random number R_t $Q = H_l(ID \parallel r_T \parallel r_R)$ $K_{i,T} = K_i \oplus r_T$ $ID = H(ID)$ If $RT(Q) \stackrel{?}{=} RT(Q')$ $K_{i+1} \leftarrow H(r_T \oplus K_i)$ *End*

Fig. 8. The improved version of the *Sun-Zhong* protocol.

Table 2. A comparison of security and privacy of protocols

Protocols / Attacks	A [16]	B [17]	C [13]	D	E [20]	F [21]	G	H [22]	I [23]	J
Secret Parameters Reveal	×	×	×	✓	✓	✓	✓	✓	✓	✓
Backward Traceability	×	✓	✓	✓	✓	✓	✓	×	✓	✓
Traceability	×	✓	×	✓	×	×	✓	×	✓	✓
Forward Traceability	✓	✓	×	✓	✓	×	✓	✓	×	✓
Impersonation	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
DoS	✓	×	✓	✓	✓	✓	✓	✓	✓	✓

✓: Secure ×: Insecure

A. Yeh et al. B. Pang et al. C. Wang et al. D. Improved Wang et al. E. Chen et al. F. Safkhani et al. G. Improved Safkhani et al. H. Ha et al. I. Sun-Zhong J. Improved Sun-Zhong

Table 3. A comparison of computational complexity

Protocols	Tag's Computational Complexity	Reader's Computational Complexity	Server's Computational Complexity
Wang et al. [13]	8 PRNG	8 PRNG	---
Improved Wang et al.	8 PRNG	8 PRNG	---
Safkhani et al. [21]	2 SE + 1 PRNG	1 SE + 1 PRNG	3 H + 1 SE
Improved Safkhani et al.	2 SE + 2 PRNG	1 SE + 1 PRNG	3 H + 1 SE
Sun-Zhong [23]	2 H + 1 PRNG	3 H + 1 PRNG	---
Improved Sun-Zhong	3 H + 1 PRNG	4 H + 1 PRNG	---
Habibi et al. [29]	6 PRNG	1 H + 1 PRNG	1 H + 7 PRNG
Improved Habibi et al. [12]	7 PRNG	1 H + 1 PRNG	1 H + 7 PRNG

H: hash function, PRNG: Pseudo Random Number Generator, SE: Symmetric Encryption

6.4 Security and Efficiency

The security and privacy properties of *Yeh et al.*'s protocol [16], *Pang et al.*'s protocol [17], *Wang et al.*'s protocol [13], *Chen et al.*'s protocol [20], *Safkhani et al.*'s protocol [21], *Ha et al.*'s protocol [22], *Sun* and *Zhong*'s protocol [23] and the modified protocols are summarized in Table 2. As illustrated in Table 2, *Yeh et al.*'s protocol not only suffers from secret parameter reveal attack, but also does not provide confidential and untraceable communications for RFID end users. The mentioned attacks are reported with more details in [29]. In [13], *Wang et al.* showed that *Pang et al.*'s protocol is vulnerable to DoS attack and also the adversary can obtain secret parameters with $O(2^{16})$ attack complexity. In Section 3, we showed that *Wang et al.*'s protocol not only cannot provide users privacy, but also the secret parameters can be disclosed by $O(2^{16})$ attack complexity; the main idea of this attack is presented in [30]. As it can be seen, in the improved version of *Wang et al.*'s protocol, all the mentioned drawbacks have been eliminated and it became secure against various security and privacy attacks.

In [21], *Safkhani et al.* showed that *Chen et al.*'s protocol is insecure against traceability and impersonation attacks. Then, they proposed some modification and have tried to provide a more efficient protocol. In Section 4, it is shown that still *Safkhani et al.*'s protocol is not safe against traceability and forward traceability attacks and an adversary can trace the location of a specific tag in the current and future runs. According to the presented modifications and privacy analysis in Section 6.2, it can be seen that the modified version of *Safkhani et al.*'s protocol can protect RFID end-users against various security and privacy attacks.

Ha et al.'s protocol [22] and *Sun* and *Zhong*'s protocol [23] are two efficient hash-based RFID authentication protocols which have been proposed in the last few years. In [23], *Sun* and *Zhong* analyzed *Ha et al.*'s protocol and showed that it has some privacy weaknesses and suffers from traceability and forward traceability attacks. In Section 5.2, we illustrated that although in [23] *Sun* and *Zhong* have tried to omit privacy concerns of *Ha et al.*'s protocol [22], but still there is a privacy concern in the improved protocol and *Sun* and *Zhong*'s protocol cannot provide forward privacy. On the other hand, privacy analysis shows that improved version of *Sun* and *Zhong*'s protocol removes all privacy concerns and provides secure and confidential communications for RFID users.

Table 2 summarizes all the discussed security and privacy analysis and computational complexity of proposed schemes with respect to several related works have compared in Table 3.

7 Conclusion

Privacy providing of RFID end-users is one of the primary goals of each RFID authentication protocol. In this paper, we analyzed the privacy of three RFID authentication protocols proposed by *Wang et al.* [13], *Safkhani et al.* [21], and *Sun-Zhong* [23] in 2012, 2012 and 2014, respectively. We showed that the privacy of all the mentioned protocols has some weaknesses and we presented various traceability attacks against each one of the studied protocols. In our privacy analysis, we used the well-known formal RFID privacy model of *Ouafi* and *Phan* [28]. Moreover, in order to overcome the existing weaknesses of the studied protocols, we applied some modifications and proposed an improved version of each one. Finally, the privacy of the proposed protocols were compared with some similar protocols.

References

1. Vaudenay, S. (2007). E-passport threats. *IEEE Security & Privacy*, 5(6), 61-64.
2. Heyden, D. (2014). RFID Applications. Available: http://www.fibre2fashion.com/industry-article/11/1023/rfid-applications1.asp. [Accessed 11 February 2014].
3. Ok, M. H., & Uiwang G. (2009). A location tracking by RFID to assist the transportation vulnerable in subway stations. *11th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering*.
4. Ruiz-Garcia L., & Lunadei L. (2011). The role of RFID in agriculture: Applications, limitations and challenges. *Computers and Electronics in Agriculture* 79(1), 42-50.
5. Ng M. L., Leong K. S., Hall D. M., & Cole P. H. (2005). A small passive UHF RFID tag for livestock identification. *IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*.
6. Mishra D., Das A. K., Mukhopadhyay S., & Wazid M. (2016). A Secure and Robust Smartcard-Based Authentication Scheme for Session Initiation Protocol Using Elliptic Curve Cryptography. *Wireless Personal Communications*, 91(3), 1361-1391.
7. Avoine G. (2005). Cryptography in Radio Frequency Identification and Fair Exchange Protocols. PHD Thesis, Lausanne, University of EPFL.
8. Juels A. (2006). RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications* 24(2), 381–394.
9. Gross H., Wenger E., Martín H., & Hutter M. (2014). PIONEER: a prototype for the internet of things based on an extendable EPC Gen2 RFID tag. *Radio Frequency Identification: Security and Privacy Issues*, 54-73.
10. Hada H., & Mitsugi J. (2011). EPC based internet of things architecture. *IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*.
11. Baghery K., Abdolmaleki B., Akhbari B., Aref M. R. (2016). Enhancing privacy of recent authentication schemes for low-cost RFID systems. *The ISC International Journal of Information Security*, 7 (2), 135-149.
12. Alavi S. M., Baghery K., Abdolmaleki B., & Aref M. R. (2015). Traceability analysis of recent RFID authentication protocols. *Wireless Personal Communications*, 83(3), 1663-1682.
13. Wang S., Liu S., & Chen D. (2014). Security analysis and improvement on two RFID authentication protocols. *Wireless Personal Communications*, 82(1), 21-33.
14. Farash M. S. (2014). Cryptanalysis and improvement of an efficient mutual authentication RFID scheme based on elliptic curve cryptography. *The Journal of Supercomputing* 70(2), 987-1001.
15. Mishra D. (2016). Design and Analysis of a Provably Secure Multi-server Authentication Scheme. *Wireless Personal Communications* 86(3), 1095-1119.
16. Yeh T. C., Wanga Y. J., Kuo T. C., & Wanga S. S. (2010). Securing RFID systems conforming to EPC Class 1 Generation 2 standard. *Expert Systems with Applications*, 37, 7678–7683.
17. Pang L., He L., Pei Q., & Wang Y. (2013). Secure and efficient mutual authentication protocol for RFID conforming to the EPC C-1 G-2 Standard. *IEEE Wireless Communications and Networking Conference (WCNC)*.
18. EPCglobal Inc. Available: http://www.epcglobalinc.org
19. Amendola S., Lodato R., Manzari S., Occhiuzzi C., & Marrocco G. (2014). RFID technology for IoT-based personal healthcare in smart spaces. *IEEE Internet of Things Journal*, 1(2), 144-152.
20. Chen Y. Y., Huang D. C., Tsai M. L., & Jan J. K. (2012). A design of tamper resistant prescription RFID access control system. *Journal of Medical Systems*, 36(5), 2795-2801.

21. Safkhani M., Bagheri N., & Naderi M. (2012). On the designing of a tamper resistant prescription rfid access control system. *Journal of Medical Systems* 36(6), 3995-4004.
22. Ha J., Moon S., Zhou J., & Ha J. (2008). A new formal proof model for RFID location privacy. *Computer Security-ESORICS*.
23. Sun D. Z., & Zhong J. D. (2012). A hash-based RFID security protocol for strong privacy protection. *IEEE Transactions on Consumer Electronics* 58(4), 1246-1252.
24. Coisel I., & Martin T. (2013). Untangling RFID privacy models. *Journal of Computer Networks and Communications, DOI:10.1155/2013/710275.*
25. Avoine G. (2005). Adversarial model for radio frequency identification. *Cryptology ePrint Archive, report 2005/049. http://eprint.iacr.org/2005/049*.
26. Juels A., & Weis S. (2007). Defining strong privacy for RFID. *5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*.
27. Vaudenay S. (2007). On privacy models for RFID. *ASIACRYPT 2007, LNCS 4833*.
28. Ouafi K., & Phan R. CW. (2008). Privacy of recent RFID authentication protocols. *4th International Conference on Information Security Practice and Experience (ISPEC)*.
29. Habibi M. H., & Gardeshi. M. (2011). Cryptanalysis and improvement on a new RFID mutual authentication protocol compatible with EPC standard. *8th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC).*
30. Abdolmaleki B., Baghery K., Akhbari B. & Aref, M. R. (2015). Cryptanalysis of two EPC-based RFID security schemes. *12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), pp. 116-121.*