Game-Based Cryptanalysis of a Lightweight CRC-Based Authentication Protocol for EPC Tags

K. Baghery¹, B. Abdolmaleki¹ and M. J. Emadi²*

¹-MSc Student, Department of Electrical Engineering, Shahed University, Tehran, Iran
²- Assistant Professor, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

The term "Internet of Things (IoT)" expresses a huge network of smart and connected objects which can interact with other devices without our interposition. Radio frequency identification (RFID) is a great technology and an interesting candidate to provide communications for IoT networks, but numerous security and privacy issues need to be considered. In this paper, we analyze the security and the privacy of a new RFID authentication protocol proposed by Shi et al. in 2014. We prove that although Shi et al. have tried to present a secure and untraceable authentication protocol, their protocol still suffers from several security and privacy weaknesses which make it vulnerable to various security and privacy attacks. We present our privacy analysis based on a well-known formal privacy model which is presented by Ouafi and Phan in 2008. Moreover, to stop such attacks on the protocol and increase the performance of Shi et al.’s scheme, we present some modifications and propound an improved version of the protocol. Finally, the security and the privacy of the proposed protocol were analyzed against various attacks.

KEYWORDS

Internet of Things, RFID authentication protocols, Security and Privacy, Ouafi-Phan Privacy Model, EPC C1 G2 Standard.
1- INTRODUCTION

RFID is a user friendly technology which is useful in various applications in which identification, tracking or authentication are necessary [1]. An RFID system could be the best choice for asset management, tracking and positioning with precision, supply chain management, healthcare control, automobile ignition keys, production control and pass control [2]-[6]. Besides, RFID systems are interesting and popular candidates to be implemented in the Internet of Things world which is introduced as a next generation of internet [7]. In the IoT paradigm, we will face a huge global network which makes connections between large number of smart and IP-based devices in our environments Anytime, Anyplace, with Anything and Anyone [8]. Communications between IoT elements may be set up via various sensing devices like Global Positioning System (GPS), intelligent sensors, RFID systems or any other smart device that can exchange data between two objects [9]. Mainly, an RFID system has three main parts including back-end server, readers and large number of tags. The architecture of an RFID system is illustrated in Fig. 1. The tags are transponders equipped with a microstrip antenna and communicate with the readers using radio waves. Due to the nature of wireless communications, communication channels between the tags and the readers are not secure and can be accessed by an outsider agent. Based on the power supply, available memory, operational frequency, processing power and range of work, the tags are classified to various categories which are employed in the desired applications. The second parts of each RFID system are the readers which act as interrogators and exchange messages between the tag and back-end server. This fact is graphically shown in Fig. 1. According to the desired applications, a reader can operate as a fixed or mobile reader. In the case that the reader is mobile, wireless communication channels between the readers and the back-end server might be insecure. The third and the essential part of each RFID system is the back-end server which acts as a core of an RFID system and performs various processing such as identification and authentication of the tags and in some cases the readers. The back-end server has all secret information about the tags and utilizes them in authentication procedures [10]. Usually, the back-end server is a central computer which has a powerful Central Processor Unit (CPU) and is connected to readers over a wireless or wired channel.

In the IoT paradigm, RFID tags can be attached to different objects and make a wireless connection with the RFID readers which act as an IoT gateway. A simple communication scenario of an RFID system in the IoT world is shown in Fig. 2. According to the figure, it can be seen that RFID readers can play the role of IoT gateway. In [8], Gross et al. proposed a prototype for the IoT paradigm based on the RFID passive tags in which the tags are conforming to the Electronic Product Code Class 1 Generation 2 (EPC C1 G2) standard. The IoT presents new services in which some of them bring security and privacy concerns for end-users. Thus, implementing a secure and confidential authentication protocol between the elements of the IoT significantly decreases these concerns.

The EPC C1 G2 standard is the most famous and popular standard which has been proposed for RFID passive tags by EPC global organization [11]. In the EPC C1 G2 standard, the tags are passive which supply their required powers using electromagnetic fields of readers. The tags, which are conforming to the EPC C1 G2 standard, have some processing limitations and are not allowed to use heavy-duty encryptions as well as hash functions [12]. This type of tag uses Pseudo Random Number Generator (PRNG), Cyclic Redundancy Code (CRC) and bitwise operators to protect the stored information and transmitted data.

In recent years, due to the widespread usage of EPC C1 G2 tags in a variety of modern applications, the security and the privacy of consumers have found great importance [13]-[14]. In this context, various lightweight RFID authentication protocols have been proposed which are under EPC C1 G2 standard and have tried to ensure the security and privacy of RFID end-users [15]-[18]. An EPC-based lightweight RFID authentication protocol is a particular security scheme that is designed to provide secure and confidential authentication between the back-end server and the tags which are conforming to the EPC C1 G2 standard. Although all the mentioned protocols are designed to protect RFID users, in the literature, several drawbacks of some EPC-based RFID authentication protocols are pointed out [12], [15], [19] and [20].

Recently, Shi et al. [21] have proposed a novel CRC-based lightweight RFID authentication protocol for EPC compliant tags. In the proposed protocol, they have used CRC and PRNG functions to protect and update the exchanged messages. In their protocol, communication channel between the tag and the reader is insecure and can be eavesdropped by an adversary. On the other hand, the reader and the back-end server communicate over a secure channel. They have analyzed the security and the privacy of their protocol against lots of existing threats including eavesdropping, traceability attacks, Denial of Service (DoS) attack, replay attack and spoofing attacks. They have claimed that the protocol can protect RFID users against various security and privacy concerns [21]. However, in this paper, we cryptanalyze Shi et al.’s protocol and we prove that due to some flaws in the structure of the exchanged messages and updating
procedures, their protocol is unable to provide secure and untraceable authentication for RFID end-users and it suffers from several security and privacy attacks. More precisely, we show that Shi et al.’s protocol is vulnerable to secret parameters reveal, tag impersonation attack and also their protocol does not provide users privacy. Then, in order to prevent all the mentioned attacks and increase the performance of the Shi et al.’s protocol, we apply some modifications in the structure of the protocol and propose an improved version of it. Our security and privacy analysis show that new modifications overcome all the existing weaknesses in Shi et al.’s protocol.

The privacy of RFID authentication protocols can be studied by two different approaches: ad-hoc [22] and formal [23]-[24]. In the ad-hoc approach, the adversary defines some notation and performs an attack based on the defined notations. On the other hand, in the formal approaches, the abilities of the adversary are classified into different categories which can be used in different privacy analysis. In the different studies, several RFID formal privacy models are proposed [25]-[31]. In this paper, we use a formal RFID privacy model which proposed by Ouafi and Phan (referred as Ouafi-Phan) [28] in our privacy analysis. In Ouafi-Phan privacy model, the adversary’s abilities are classified into four categories including Execute Query, Corrupt Query, Sent Query and Test Query which are discussed with more details in the following section.

The rest of this paper is organized as follows. In section 2, Ouafi-Phan formal privacy model is described. We review Shi et al.’s protocol in section 3. Security and privacy weaknesses of Shi et al.’s protocol are investigated in section 4. In section 5, an improved and robust version of Shi et al.’s protocol is proposed. In this section, the security and the privacy of the proposed protocol are also analyzed and compared with some similar protocols. The paper is concluded in Section 6.

2-OUAFI AND PHAN PRIVACY MODEL

In 2008, Ouafi and Phan [28] presented a privacy model to evaluate RFID authentication protocols. In this paper, we use this model for our privacy analysis. So, in this section, we summarize Ouafi-Phan privacy model which will be used in the rest of paper.

In this model, the adversary $\mathcal{A}$ can eavesdrop all channels between tags and readers and also it can attack them actively or passively. Similarly, the adversary $\mathcal{A}$ has been allowed to run the following queries:

- **Execute query ($R, T, i$)**: Passive attacks take place in this query. In other words, the adversary can eavesdrop all transmitted messages between the tag $T$ and the reader $R$ in $i$th session. As a result, the adversary 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 adversary $\mathcal{A}$ has permission to impersonate a reader $U$ in the $i$th session, and forwards a message $m$ to a tag $V$. In addition, the adversary $\mathcal{A}$ has permission to alert or block the exchanged message $m$ between the tag and the reader. Note that $U$ and $V$ are the members of readers and tags sets, respectively.

- **Corrupt query ($T, K'$)**: In this query, the adversary $\mathcal{A}$ has permission to access secret keys of the tag. In fact, the adversary $\mathcal{A}$ has physical access to the tag’s database. In addition, the adversary $\mathcal{A}$ 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 $i$th session, a random number bit $b \in \{0,1\}$ is generated by challenger and delivered $T_b \in \{T_0, T_1\}$ to the adversary. Now, the adversary 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 adversary $\mathcal{A}$ and a set of the tag and the reader instances. In other words, an adversary $\mathcal{A}$ 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 adversary $\mathcal{A}$ has permission to send each one of the queries such as Execute, Send

Fig. 2. A communication scenario of RFID tags and readers in the IoT world.
and Corrupt, and interact with the reader R and T₀, T₁ that are chosen randomly.

- **Challenge phase:** The adversary A selects two tags T₀, T₁ and forwards a Test query (T₀, T₁, i) to the challenger. After that, the challenger selects b ∈ {0, 1} randomly and the adversary A determines a tag T_b ∈ {T₀, T₁} using Execute and Send queries.

- **Guess phase:** Eventually, the adversary A finishes the game G and outputs a bit b* ∈ {0, 1} as guess of b. The success of adversary A in game G and consequently breaking the notion of UPriv is quantified via A’s advantage in recognizing whether adversary A received T₀ or T₁, and denoted by Adv_{UPriv}^A(k) where k is the security parameter.

\[
Adv_{UPriv}^A(k) = \left| \Pr(b^* = b) - \Pr(\text{random coin flip}) \right|
\]

Where 0 ≤ Adv_{UPriv}^A(k) ≤ 1/2. Note that, if Adv_{UPriv}^A(k) ≪ ε(k), the protocol is traceable with a negligible probability.

3-SHI ET AL.’S PROTOCOL

Recently, in [21], Shi et al. presented a five-step CRC-based authentication protocol for RFID systems. The notations used in the paper are presented in Table 1.

### TABLE 1. THE NOTATIONS.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>the unique identifier of a specific tag</td>
</tr>
<tr>
<td>Meta-ID</td>
<td>the pseudonym of the tag</td>
</tr>
<tr>
<td>K</td>
<td>the 32-bit secret key shared by readers and tags</td>
</tr>
<tr>
<td>CRC(<em>,</em>)</td>
<td>the CRC function</td>
</tr>
<tr>
<td>R_r</td>
<td>the pseudorandom number generated by a reader</td>
</tr>
<tr>
<td>R_t</td>
<td>the pseudorandom number generated by a tag</td>
</tr>
<tr>
<td>funh(x)</td>
<td>the function to get the left half-part of x</td>
</tr>
<tr>
<td>funl(x)</td>
<td>the function to get the right half-part of x</td>
</tr>
<tr>
<td>PRNG(<em>,</em>)</td>
<td>Pseudo random number generator</td>
</tr>
</tbody>
</table>

|| Concatenation operation
| A ⊕ B | Message A is XORed with message B |
| A | B | Compare whether A is equal to B or not |

The authentication procedure of Shi et al. protocol is summarized in Fig. 3 and discussed in details in a 5-step round in the following:

**Step 1.** [Reader → Tag]: The reader generates R_r as a random number and computes message M₁ = funh(k) ⊕ R_r. It then sends a Query and message M₁ to the tag.

**Step 2.** [Tag → Reader]: After receiving the reader’s response, the tag calculates R_t = M₁ ⊕ funh(k), then it generates a random number R_t and computes the following messages and sends them to the reader.

\[
M_2 = CRC(funl(Meta_ID) ⊕ R_t) \parallel CRC(funh(Meta_ID) ⊕ R_r).
\]

**Step 3.** [Reader → Back-end server]: By using the message M₄, the reader abstracts the random number R₄, and then it forwards messages (M₂, M₃, R₄, R₁) to the back-end server.

**Step 4.** [Back-end server → Reader]: Upon receiving the sent messages from the reader, the back-end server performs the following operations.

\[a)\] Using new_ID and new_Meta_ID or old_ID and old_Meta_ID, it generates M₅ = CRC(funh(X_Meta_ID) ⊕ R₄) \parallel CRC(funh(X_Meta_ID) ⊕ R₄), and M₃ = CRC(funh(X_ID) ⊕ R₄ ⊕ R₄) for X = old and new. Afterwards, it verifies M₅ = M₃ \parallel M₃ and determines that X = old or new. If M₃ = M₃ and M₃ = M₃ for X = old or new, it authenticates the tag and responds to the reader through the following messages.

\[
M_5 = CRC(funh(Meta_ID) ⊕ funl(ID) ⊕ R_t),
\]

\[
M_6 = CRC(funh(Meta_ID) ⊕ funh(ID) ⊕ R_r).
\]

Otherwise, the back-end server quit the protocol.

**Step 5.** [Reader → Tag]: The reader sends message M₅ and M₆ to the tag. Upon receiving messages from the reader, the tag calculates message M₅ and M₆, then in order to authenticate the back-end server, the tag verifies M₅ = M₅ and M₆ = M₆. Finally it updates its secret values as

\[
\text{new_ID} ← \text{PRNG}(\text{funh(new_ID) } \oplus \text{ R₄}) \parallel \text{PRNG}(\text{funh(new_ID) } \oplus \text{ R₄})
\]

\[
\text{old_Meta_ID} ← \text{PRNG}(\text{funh(new_Meta_ID)} \oplus \text{ R₄}) \parallel \text{PRNG}(\text{funh(new_Meta_ID)} \oplus \text{ R₄})
\]

Else
do nothing
End
For each Meta_ID and ID in DB, the server generates M1 and M2 to verify the tag and determines X = old or new for Meta_ID and ID. If X = new, the server acts as follows:

\[ M_1 = CRC(funh(Meta_ID) \oplus funh(ID) \oplus R_i) \]
\[ M_2 = CRC(funh(ID) \oplus R_i) \]

Finally, the back-end server updates its secret values as follows:

\[ R_i = R_{i} \oplus funl(R_i) \oplus R_i \]
\[ (M_1, M_2) \rightarrow \]

### 4-CRYPTOANALYSIS OF SHI ET AL.’S PROTOCOL

In [21], Shi et al. analyzed their protocol and claimed that their protocol is secure against various security and privacy attacks. We show that Shi et al.’s protocol not only cannot protect the secret keys properly but also it is vulnerable to tag impersonation and traceability attacks.

#### Linear Property

The property of CRC operator indicates that \( CRC(A \oplus B) = CRC(A) \oplus CRC(A) \), where A and B represent the arbitrary values.

#### 4-1- TAG IMPERSONATION ATTACK

In this subsection, it is shown that an adversary is able to impersonate the legitimate tag. This attack consists of two phases: learning phase and attack phase.

**Learning phase:** In the round i, the adversary acts as an eavesdropper. After one successful run, the adversary saves the exchanged data between the target tag and the reader including

\[ M_{1,i} = funh(k) \oplus R_{r,i} \]
\[ M_{2,i} = CRC(funh(Meta_ID_i) \oplus R_{r,i}) \]
\[ R_{r,i} = CRC(funh(ID_i) \oplus R_{r,i} \oplus R_{r,i}) \]
\[ M_{4,i} = funh(k) \oplus R_{r,i} \oplus R_{r,i} \]

After that, using message \( M_{2,i} \), the adversary defines \( \rho = CRC(funh(Meta_ID_i) \oplus R_{r,i}) \) and \( \varphi = CRC(funh(Meta_ID_i) \oplus R_{r,i}) \).

**Attack phase:** In this phase, the adversary acts as a legitimate tag and when the reader sends a Query to the target tag. The adversary obtains message \( M_{1,i+1} \). Then, by using obtained messages in the learning phase, the following messages are computed and sent to the reader.

\[ M_{2,att} = \rho \oplus (\varphi \oplus CRC(M_{1,i}) \oplus CRC(M_{1,i+1})) \]
\[ M_{3,att} = M_{3,i} \oplus CRC(M_{1,i}) \oplus CRC(M_{1,i+1}) \]
\[ M_{4,att} = M_{4,i} \oplus M_{1,i} \oplus M_{1,i+1} \]

Based on the receiving messages from the adversary, the reader first calculates \( R_{r,i} \) as \( R_{r,i} = M_{x,att} \oplus funh(k) \oplus R_{r,i+1} \). Then, the reader sends messages \( M_{2,att}, M_{3,att}, M_{4,att} \) and \( R_{r,i+1} \) to the back-end server. To verify the adversary as a legitimate tag, using old ID and Meta_ID, the back-end server performs two phases as follows:

**Phase 1:** First the adversary calculates message \( M_{2}' \) as

\[ M_{2}' = CRC(funh(old_Meta_ID) \oplus R_{r,i}) \]

\[ CRC(funh(old_Meta_ID) \oplus R_{r,i+1}) \]

and verifies \( M_{2}' \) as follows,

\[ M_{2,att} = \rho \oplus (\varphi \oplus CRC(M_{1,i}) \oplus CRC(M_{1,i+1})) \]

By using the linear property, we have

\[ M_{2,att} = \rho \oplus (\varphi \oplus CRC(funh(k)) \oplus CRC(R_{r,i})) \]
\[ \oplus CRC(funh(k) \oplus R_{r,i+1}) \]

Then, by substituting \( \varphi = CRC(funh(Meta_ID_i) \oplus R_{r,i}) \) in equation (1), we have

\[ M_{2,att} = \rho \oplus (CRC(funh(Meta_ID_i) \oplus R_{r,i}) \oplus CRC(R_{r,i+1}) \oplus CRC(R_{r,i})) \]

Again, by considering the linear property, \( M_{2,att} \) is
rewritten as
\[ \rho \| \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus \text{CRC}(R_{t,i}) \right) \]
\[ \quad \oplus \left( \text{CRC}(\text{MetaID}) \oplus \text{CRC}(R_{t,i+1}) \right) \]
\[ = \rho \| \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus \text{CRC}(R_{t,i}) \right) \]
\[ \quad \oplus \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus \text{CRC}(R_{t,i+1}) \right) \]
\[ = \rho \| \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus R_{t,i+1} \right). \quad (3) \]

Finally, by substituting \( \rho = \text{CRC}(\text{funl}(\text{MetaID})) \oplus R_{t,i} \) in equation (3), we can write
\[ = \text{CRC}(\text{funl}(\text{MetaID})) \oplus R_{t,i} \]
\[ \quad \| \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus R_{t,i+1} \right) \]
\[ = \text{CRC}(\text{funl}(\text{old MetaID})) \oplus R_{t,i} \]
\[ \quad \| \left( \text{CRC}(\text{funh}(\text{MetaID})) \oplus R_{t,i+1} \right) \]
\[ = M'_2. \quad (4) \]

**Phase 2:** The back-end sever computes message \( M'_3 \) as \( M'_3 = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus R_{t,i+1} \). Then, in order to authenticate the adversary as a legitimate tag, back-end sever verifies \( M'_3 \equiv M_{3,\text{att}} \) as follows:
\[ M_{3,\text{att}} = M_{3,i} \oplus \text{CRC}(M_{1,i}) \oplus \text{CRC}(M_{1,i+1}) \]

Substituting \( M_{3,i} = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus R_{t,i+1} \), \( M_{1,i} = \text{funh}(k) \oplus R_{t,i} \) and \( M_{1,i+1} = \text{funh}(k) \oplus R_{t,i+1} \), equation (4) can be rewritten as follows,
\[ M_{3,\text{att}} = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus R_{t,i+1} \]  
\[ \oplus \text{CRC}(\text{funh}(k) \oplus R_{t,i} \oplus R_{t,i+1}). \quad (5) \]

By using the linear property of CRC operation, we have
\[ = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus \text{CRC}(R_{t,i}) \]
\[ \oplus \text{CRC}(\text{funh}(k) \oplus \text{CRC}(R_{t,i}) \oplus \text{CRC}(\text{funh}(k)) \oplus \text{CRC}(R_{t,i+1}) \]
\[ = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus \text{CRC}(R_{t,i+1}) \]
\[ = \text{CRC}(\text{funl}(\text{ID})) \oplus R_{t,i} \oplus R_{t,i+1} \]
\[ = M'_3. \quad (6) \]

Therefore, the back-end server authenticates the adversary as a legitimate tag.

### 4-2: SECRET PARAMETER REVEAL ATTACK

In this subsection, we present a practical secret parameter reveal attack against Shi et al.’s protocol. It is shown that an adversary is able to reveal secret parameter MetaID and ID. This attack is performed in two phases as follows.

**Learning phase:** In this phase, the adversary acts as an eavesdropper. After two successful runs of the protocol, the adversary saves the exchanged data between the target tag and the reader including
\[ M_{2,i} = \text{CRC}(\text{funl}(\text{MetaID})) \oplus R_{t,i} \]
\[ \text{CRC}(\text{funh}(\text{MetaID}) \oplus R_{t,i}) \]
\[ M_{2,i+1} = \text{CRC}(\text{funl}(\text{MetaID})_{i+1}) \oplus R_{t,i+1} \]
\[ \text{CRC}(\text{funh}(\text{MetaID})_{i+1}) \oplus R_{t,i+1} \]
\[ M_{5,i+1} = \text{CRC}(\text{funh}(\text{MetaID})_{i+1}) \oplus \text{funl}(\text{ID})_{i+1} \]
\[ \text{funh}(\text{MetaID})_{i+1} \oplus R_{t,i+1} \] and
\[ M_{6,i+1} = \text{CRC}(\text{funl}(\text{MetaID})_{i+1}) \]
\[ \text{funh}(\text{MetaID})_{i+1} \oplus R_{t,i+1} \]

**Attack phase:** The adversary defines two new parameters \( \rho \) and \( \varphi \) as \( \rho = \text{CRC}(\text{funl}(\text{MetaID}) \oplus R_{t,i}) \). \( \varphi = \text{CRC}(\text{funh}(\text{MetaID}) \oplus R_{t,i}) \) which are the first and the second parts of message \( M_{2,i} \). Then adversary performs the following steps;

a) Since \( (\text{funl}(\text{MetaID}) \oplus R_{t,i}) \) is a 16-bit string, thus \( (\text{funl}(\text{MetaID}) \oplus R_{t,i}) \in U \) where \( U = \{ u_1, u_2, ..., u_{2^{16}} \} \). Now, using the new parameter \( \rho \),

\[ \text{For} \ 1 \leq j \leq 2^{16} \]
\[ \text{Choose} \ u_j \in U \]
\[ \text{If} \ \rho = \text{CRC}(u_j) \]
\[ \text{return} \ u_j \text{ as } (\text{funl}(\text{MetaID}) \oplus R_{t,i}) \]
\[ \text{End} \]

b) Now, like step (a), since \( (\text{funh}(\text{MetaID}) \oplus R_{t,i}) \) is a 16-bit string, thus \( (\text{funh}(\text{MetaID}) \oplus R_{t,i}) \in V \) where \( V = \{ v_1, v_2, ..., v_{2^{16}} \} \). Now, using the new parameter \( \varphi \),

\[ \text{For} \ 1 \leq j \leq 2^{16} \]
\[ \text{Choose} \ u_j \in U \]
\[ \text{If} \ \varphi = \text{CRC}(u_j) \]
\[ \text{return} \ u_j \text{ as } (\text{funh}(\text{MetaID}) \oplus R_{t,i}) \]
\[ \text{End} \]

Now, by using \( (\text{funl}(\text{MetaID}) \oplus R_{t,i}) \) and \( \text{funh}(\text{MetaID}) \oplus R_{t,i} \) in the steps (a) and (b), the adversary calculates the secret value MetaID_{i+1} as
\[ \text{MetaID}_{i+1} = \text{PRNG}(\text{funl}(\text{MetaID}) \oplus R_{t,i}) \]
\[ \text{PRNG}(\text{funh}(\text{MetaID}) \oplus R_{t,i}) \]
that will be used in the round \( (i + 1) \).

c) In order to compute the secret value \( K \), the adversary uses the eavesdropped messages \( M_{2,i+1}, M_{3,i+1} \) and \( M_{4,i+1} \) in the learning phase and the linear property of CRC operator, adversary calculates \( R_{t,i+1} \) and \( R_{t,i+1} \) as follows.

- First, in order to calculate \( R_{t,i+1} \), adversary uses the first part of the message \( M_{2,i+1} \) and calculates the secret value MetaID_{i+1} in steps (a) and (b), so
\[ R_{t,i+1} = \text{CRC}(\text{funl}(\text{MetaID})_{i+1}) \]
\[ \oplus \text{CRC}(\text{funl}(\text{MetaID})_{i+1}) \]

Using the linear property, \( R_{t,i+1} \) is rewritten as
\[ = \text{CRC} \left( \text{funl}(\text{MetaID})_{i+1} \right) \]
**Proof:** According to Shi et al.’s protocol, the following equations are given

\[
\rho^{T_b} \oplus \varphi^{T_b} \oplus M^T_{i,j+1} = CRC(fun(Meta_{ID}^{T_b})) \\
\oplus R^{T_b}_{i,j+1} \oplus CRC(fun(Meta_{ID}^{T_b})) \oplus R_{r,i+1} \oplus CRC(fun(ID_{i}^{T_b})) \oplus R_{r,i+1}.
\]

Using the linear property, we have

\[
= CRC \left( fun(Meta_{ID}^{T_{b+1}}) \right) \oplus CRC \left( R^{T_{b+1}}_{i,j+1} \right) \oplus CRC \left( fun(Meta_{ID}^{T_{b+1}}) \right) \oplus CRC \left( R_{r,i+1} \right) \oplus CRC \left( R_{r,i+1} \right) \oplus CRC \left( fun(ID_{i}^{T_{b+1}}) \right) \oplus CRC \left( fun(ID_{i}^{T_{b+1}}) \right).
\]

Using this fact that \( T_b = T_0 \), we have

\[
= CRC(fun(Meta_{ID})^{T_0}) \oplus CRC(fun(Meta_{ID})^{T_0}) \oplus CRC(fun(ID_{i}^{T_0})).
\]

In the learning phase, since the tag \( T_0 \) did not update its secret values, so Meta_{ID}^{T_0} = Meta_{ID}^{T_0} and \( ID_{i}^{T_0} = ID_{i}^{T_0} \), as a result

\[
= CRC(fun(Meta_{ID})^{T_0}) \oplus CRC(fun(ID_{i}^{T_0}) \oplus CRC(fun(ID_{i}^{T_0}) \oplus CRC(fun(ID_{i}^{T_0}) \oplus CRC(fun(ID_{i}^{T_0}) = \xi.
\]

In summary, we proved that an adversary can trace the location of a specific tag in a specific session.

**5- IMPROVED VERSION OF SHI ET AL.’S PROTOCOL**

In this section, we propose some modifications in the structure of Shi et al.’s protocol to overcome all the reported weaknesses in Section 4. It is shown that due to some flaws in the tag responses and updating procedure of the Shi et al.’s protocol, their protocol cannot protect RFID users against secret parameter reveal, impersonation.
and traceability attack. Thus, in the improved version, we propose some changes in the exchanges messages between the tag and the reader, and modify the updating procedure of the tag and the back-end server. The changes and modifications are discussed in details in the following.

- In Shi et al.’s protocol, the values of $M_2$ and $M_3$ are given by $M_2 = \text{CRC}(\text{funh(Meta}_{ID}) \oplus R_t)$ and $M_3 = \text{CRC}(\text{funh(ID)} \oplus R_t \oplus R_r)$. We change their values to $M_2 = \text{PRNG}(\text{funh(Meta}_{ID}) \oplus R_t)$ and $M_3 = \text{PRNG}(\text{funh(ID)} \oplus R_t \oplus R_r)$.
- The next change is in updating the tag and the back-end server as follows:

\[
\begin{align*}
\text{old}_{ID} &\leftarrow \text{new}_{ID} \leftarrow \text{CRC}(\text{funh(new}_{ID}) \oplus R_r) \mid \text{CRC}(\text{funh(new}_{ID}) \oplus R_r). \\
\text{old}_{\text{Meta}_{ID}} &\leftarrow \text{new}_{\text{Meta}_{ID}} \leftarrow \text{CRC}(\text{funh(new}_{\text{Meta}_{ID}}) \oplus R_r) \mid \text{CRC}(\text{funh(new}_{\text{Meta}_{ID}}) \oplus R_r).
\end{align*}
\]

All authentication steps of the improved protocol are the same as Shi et al.’s protocol, except the proposed modifications in the updating procedure and the tag responses. Final structure of the improved protocol is shown on Fig. 4 wherein the authentication steps are provided with more details.

In the rest of this section, it is shown that how these changes prevent all the presented attacks and make the protocol more efficient and robust than before.

### 5-1. SECRET PARAMETER REVEAL

As it is shown in subsection 4-1, due to the dependency between the updating of secret keys and the structure of the tag response $M_2$, Shi et al.’s protocol cannot protect secret keys and an adversary can obtain the secret parameters with maximum $2^{16}$ computations. In the improved protocol, this problem is eliminated with our new changes in the updating procedure of Meta$_{ID}$ and $M_3$ structure.

### 5-2. IMPersonation AND REplay ATTACK

In the proposed improved version of Shi et al.’s protocol, due to some changes applied in messages $M_2 = \text{PRNG}(\text{funh(Meta}_{ID}) \oplus R_t)$ and $M_3 = \text{PRNG}(\text{funh(ID)} \oplus R_t \oplus R_r)$, which are exchanged between the tag and the reader, by using PRNG operator instead of CRC operator, the weaknesses that are reported in section 4 are omitted. Therefore, the adversary cannot use the eavesdropped messages and perform impersonation and replay attack.

#### 5-3. PRIVACY

Providing confidential and untraceable communications for the end-users is one of the main goals of each RFID authentication protocol. In subsection 4-3, we showed that the privacy of Shi et al.’s protocol has some drawbacks and makes it unable to provide untraceable communication. In the modified protocol, we solve this problem by changing the message $M_3$ as $M_3 = \text{PRNG}(\text{funh(ID)} \oplus R_t \oplus R_r)$ and updating of Meta$_{ID}$ as $\text{Meta}_{ID} = \text{CRC}(\text{funh(Meta}_{ID}) \oplus R_r)$ and $\text{CRC}(\text{funh}(\text{Meta}_{ID}) \oplus R_r)$. With these modifications,
an adversary cannot remove the effect of random numbers $R_t$ and $R_r$ and traces the location of a specific tag.

Finally, we compare the security and the privacy of the improved protocol with some similar new-found RFID authentication protocols in Table 2. According to the last column, it can be seen that all the discovered drawbacks are eliminated in the improved version.

### TABLE 2. A COMPARISON OF SECURITY ANALYSIS.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret Values</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✔</td>
</tr>
<tr>
<td>Reveal</td>
<td>×</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Replay</td>
<td>×</td>
<td>✔</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Impersonation</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>DoS</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Traceability</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

* ✔ Secure ×: Insecure


### 5- CONCLUSION

We cryptanalyzed a CRC-based lightweight mutual authentication protocol which has been proposed recently for RFID systems by Shi et al. [21]. Shi et al. claimed that their protocol is safe against different security and privacy attacks. However, we showed that their protocol has some drawbacks which make it vulnerable to secret parameter reveal, tag impersonation and traceability attacks. We presented our traceability attack based on a well-known RFID formal privacy model proposed by Ouafi and Phan. Moreover, in order to increase the performance of Shi et al.’s protocol and prevent the presented attacks, we proposed some modifications in the structure of the original protocol and presented an improved protocol which removes all the existing weaknesses. The analysis illustrated that the improved protocol can provide secure and untraceable communication for RFID end-users. Finally, a comparison of security analysis for the improved protocol and some similar RFID authentication protocols was presented.

### REFERENCE


