ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 2, No.8 , March 2014
ISSN : 2322-5157
www.ACSIJ.org

Analysis and Improvement of the Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard

Masoud Mohammadi[1*], Mehdi Hosseinzadeh[2] and Mohammad Esmaeildoust [3]

[1*] Department of Computer Engineering, Khouzestan Science and Research Branch, Islamic Azad University, Ahvaz, Iran
M28_Mohammadi@yahoo.com

[2] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
Hosseinzadeh@srbiau.ac.ir

[3] Department of Marine Engineering, Marine Science and Technology University of Khorramshahr, Iran
M_Doust@kmsu.ac.ir

Abstract

Radio Frequency Identification (RFID) technology is a promising technology. It uses radio waves to identify objects. Through automatic and real-time data acquisition, this technology can give a great benefit to various industries by improving the efficiency of their operations. However, this ubiquitous technology has inherited problems in security and privacy. EPC Class 1 Generation 2 has served as the most popular standard for passive RFID tags. To improve the security of this standard, several protocols have been proposed compliant to this standard. In this paper we analyze the revised Yeh et al.'s(2010) protocol by Habibi et al.'s(2011) which is conforming to EPC-C1 G2 standard and is one of the most recent proposed protocols in this field. We discuss several drawbacks of this protocol, then we present our enhanced protocol which the security analysis showed that it can improve the security and privacy of RFID systems.

Keywords: RFID, EPC, Mutual Authentication, Security, Privacy, Adversary

1. Introduction

Radio Frequency Identification, abbreviated ''RFID'' basically provides a means to identify objects having RFID tags attached. Fundamentally, RFID tags provide the same functionality as barcodes but usually have a globally unique identifier. Using RFID, the identification is performed electromagnetically. Thus, there is, in contrast to barcodes, no line-of-sight necessary, and the identification can also be performed in contactless way. RFID also has the advantage that bulk reading is possible and that it is not susceptible to dust, dirt, or vibration like barcodes. Because of these characteristics, RFID is envisioned to be a convenient replacement for optical barcodes in the future [1].

There are several interconnected standards for RFID systems. Among them, ISO and EPC global have played the main role. In 2004 [2,3], the Electronic Product Code Class-1 Generation-2 specification (EPC-C1 G2 in short) was announced by EPC Global which also has been ratified by ISO [4] and published as an amendment to ISO/IEC18000-6. This standard is an important milestone for the standardization of low-cost RFID tags. However, the later security analysis that carried out on the EPC-C1 G2 specification have demonstrated important security flaws in this standard [5,6]. This is motivated researchers to try to propose EPC-compliant schemes, trying to correct the weaknesses and improve its security level, analyze the security of EPC-compliant schemes, or improve the vulnerable schemes [7,8,9,10,11,12,13,14]. Among them, one of the most recent proposals that following this approach is an improvement to the Yeh et al. 's protocol [14] proposed by Habibi et al. [12], which is the main concern of this paper. Habibi et al. [12] have analyzed the security of Yeh et al.'s protocol and proposed an improved protocol as a treatment for Yeh et al.'s protocol. However, other researches [7,11] have demonstrate that they were not success in their attempt and the proposed protocol has security and privacy problems. In this paper we proposed an enhanced protocol that improving cited problems. The security analysis showed that the proposed protocol can improve the security and privacy of RFID systems. Also, it can be applied in low-cost RFID environments requiring a high level of security.

The remaining sections of the paper are organized as follows: Section2 briefly reviews Habibi et al.'s protocol. Section3 discusses the weakness of Habibi et al.'s protocol. The enhanced protocol is presented in Section4, while Section5 discusses the security analysis of the proposed protocol, respectively. Some conclusions are presented in Section6.

2. Review of Habibi et al.'s protocol

This section reviews Habibi et al.'s protocol [11].

Notations used in this paper are defined as follows:

- EPCs: The 96 bits of EPC code are divided into six 16-bit blocks, and then the six blocks are XORed to get EPCs.
- DATA: The corresponding record for the tag kept in the database.
- K_i: The authentication key stored in the tag for the database to authenticate the tag at the (i+1)th authentication phase.
- P_i: The access key stored in the tag for the tag to authenticate the database at the (i+1)th authentication phase.
- K_{old}: The old authentication key stored in the database.
- K_{new}: The new authentication key stored in the database.
- P_{old}: The old access key stored in the database.
- P_{new}: The new access key stored in the database.
- C_i: The database index stored in the tag to find the corresponding record of the tag in the database.
- C_{old}: The old database index stored in the database.
- C_{new}: The new database index stored in the database.
- X: The value kept as either new or old to show which key in the record of the database is found matched with the one of the tag.
- $A \rightarrow B$:A forwards a message to B.
- $A \oplus B$: Message A is XORed with message B.
- RID: The reader identification number.
- H(.): Hash function.

The information kept within respective devices:

Tag: (K_i,P_i,C_i, EPC_S)

Reader: RID
DataBase:
(K_{old},P_{old},C_{old},K_{new},P_{new},C_{new},RID,EPCs,DATA)
Habibi et al.'s protocol consists of two phases: the initialization phase, and the (i+1)th authentication phase.

2.1. Initialization phase

The manufacturer generates random values for K0, P0 and C0 respectively, and sets the values for the record in the tag (K_i=K_0, P_i=P_0, C_i=C_0) and the corresponding record in the database (K_{old}=K_{new}=K_0, P_{old}=P_{new}=P_0, C_{old}=C_{new}= 0).

2.2. The (i +1)th authentication phase

The detailed steps of the authentication phase of Habibi et al.'s protocol are presented as follows:
1) The reader R generates a random number N_R and sends it to the tag T.
2) T receives N_R, generates a random number N_T, computes M1, D, E and finally sends
M1, D, E and C_i to R, where M1 = PRNG($EPC_S \oplus N_R \oplus N_T$) $\oplus K_i$ and D = $N_T \oplus K_i$ and E = $N_T \oplus$ PRNG($C_i \oplus K_i$).
3) When R receives the message, it computes V= h(RID $\oplus N_R$) and forwards M1, D, C_i, E, N_R, V to the back-end server S.
4) After S receiving M1, D, C_i, E, N_R, and V, it proceeds as follows.

- For each RID stored in the database, it computes h(RID$\oplus N_R$) and compares it with the received V to verifies R legitimacy.
- If C_i = 0, which means that it is the first access to the tag, it proceeds as follows, iteratively:
  (a) Picks up an entry (K_{old} , P_{old} , C_{old} , K_{new} , P_{new} , C_{new} , RID, EPCs, DATA) stored in database.
  (b)Verifies whether M1 $\oplus K_{old}$ = PRNG($EPC_S \oplus N_R \oplus D \oplus K_{old}$) or M1 $\oplus K_{new}$ = PRNG($EPC_S \oplus N_R \oplus D \oplus K_{new}$), and marks X as old or new provided that the verification process is satisfied based on the new record or the old record.
- Otherwise, S uses C_i as an index to find the corresponding record in the database
  and verify whether PRNG($EPC_S \oplus N_R \oplus D \oplus K_X$) $\oplus K_X$ = M1. If "No" the protocol aborts.
- Verify whether $N_T \oplus$ PRNG($C_i \oplus K_X$) = E. If "No" the protocol aborts.
- Computes M2 and Info and forwards them to R, where M2 = PRNG($EPC_S \oplus N_T$) $\oplus P_X$
  and Info = DATA $\oplus$ RID.
- If X=new, updates the database as follows:
  $K_{old} \leftarrow K_{new}$, $K_{new} \leftarrow$PRNG(K_{new}),
  $P_{old} \leftarrow P_{new}$, $P_{new} \leftarrow$PRNG(P_{new}),
  $C_{old} \leftarrow C_{new}$, $C_{new} \leftarrow$PRNG($N_T \oplus N_R$).
- Else
  $C_{new} \leftarrow$PRNG($N_T \oplus N_R$).

5) Once R receives the message, it extracts DATA as Info$\oplus$RID and forwards M2 to T.
6) When T receives the message, it verifies whether PRNG(EPCs$\oplus N_T$)=M2$\oplus P_i$.

If "No" the protocol aborts. Else T authenticates S and updates the contents kept inside as $K_{i+1} \leftarrow PRNG(K_i)$, $P_{i+1} \leftarrow PRNG(P_i)$, $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$.

3. Weaknesses of Habibi et al.'s protocol

3.1. Secret Information Disclosure Attack

Castro et al. [7] present an efficient and passive attack that retrieves any Secret Information of the tag include $EPCs$, K_i, and P_i. The adversary acts as follows:

1. Eavesdrops one session of protocol and stores all transferred messages include: N_R, C_i, $M1 = PRNG(EPC_S \oplus N_R \oplus N_T) \oplus K_i$, $D = N_T \oplus K_i$, $E = N_T \oplus PRNG(C_i \oplus K_i)$, $M2 = PRNG(EPC_S \oplus N_T) \oplus P_X$.
2. $\forall$ i=0...N_d does as follows:
   – $K_i \leftarrow i$,
   – $N_T \leftarrow D \oplus K_i$,
   – If $E = N_T \oplus PRNG(C_i \oplus K_i)$ then returns K_i and N_T.
3. For the returned value of K_i and N_T from Step2 and $\forall$ i = 0 . . . N_d does as follows:
   – EPC s ← i,
   – If $M1 = PRNG(EPC_S \oplus N_R \oplus N_T) \oplus K_i$ then returns EPC_S.
4. For the returned value of K_i and N_T from Step2 and EPCs from Step3 and $\forall$ i = 0...N_d does as follows:
   – $P_X \leftarrow i$,
   – If $M2 = PRNG(EPC_S \oplus N_T) \oplus P_X$ then returns P_X.
5. Returns the following values:
   $P_{old} = P_i$, $P_{new} = PRNG(P_i)$, $K_{old} = K_i$, $K_{new} = PRNG(K_i)$, $C_{old} = C_i$ [7].

3.2. Tag Impersonation Attack

Tag impersonation attack is a forgery attack that leads to the identification of spoofed tags by a legitimate reader. In 2012 Castro et al. [7], have shown how an adversary can deceive the reader to authenticate it as a legitimate tag. In the given tag impersonation attack, the adversary, which is an active adversary, can fallow the steps that describe bellow:

Phase1 (Learning): The adversary eavesdrops one successful run of the protocol and stores the messages exchanged between the reader and the legitimate tag including N_R, M1, D, C_i and E. At the end of this phase the records linked to this tag in the back-end database include (K_{old}, P_{old}, C_{old}, K_{new}, P_{new}, C_{new}, RID, EPSs, DATA) and the tag record includes (K_{new}, P_{new}, C_{new}, EPSs), where: $K_{new} = PRNG(K_{old})$, $P_{new} = PRNG(P_{old})$, $C_{new} = PRNG(N_T \oplus N_R)$, $M1 = PRNG(EPSs \oplus N_R \oplus N_T) \oplus K_{old}$, $D = N_T \oplus K_{old}$ and $E = N_T \oplus PRNG(C_{old} \oplus K_{old})$.

Phase 2 (Impersonation): To impersonate the legitimate tag, the adversary waits until the reader initiates a new protocol session, where:

1. The reader generates a random number N_R' and sends it to the tag.
2. After receiving N_R', the adversary replies with $M1'$, D', C_i' and E' where:
   $M1' = M1 = PRNG(EPCs \oplus N_R \oplus N_T) \oplus K_{old}$
   $C_i' = C_{old}$
   $D' = D \oplus N_R \oplus N_R' = N_T \oplus K_{old} \oplus N_R \oplus N_R'$
   $E' = E \oplus N_R \oplus N_R' = N_T \oplus PRNG(C_{old} \oplus K_{old}) \oplus N_R \oplus N_R'$
3. Once the reader receives the message, it computes $V = H(RID \oplus N_R')$ and forwards $M1'$, D', C_i', E', N_R and V to the back-end database.
4. Once the back-end database receives the message, it proceeds as follows:
   - For each stored RID in the database, computes $H(RID \oplus N_R')$ and compares it with the received V. Since the adversary has not manipulated the exchanged message from the reader to the back-end database, the back-end database authenticates the reader.
   - Assume that $C_i' \neq 0$, then back-end database uses $C_i' = C_i$ as an index to find the corresponding record in the database. The record would be found in its records for the field C_{old}. Therefore the back-end database marks X as old.
   - Verifies whether $PRNG(EPC_S \oplus N_R' \oplus D' \oplus K_{old}') \oplus K_{old} = M1$, where:
     $PRNG(EPC_S \oplus N_R' \oplus D' \oplus K_{old}) \oplus K_{old} = PRNG(EPC_S \oplus N_R' \oplus D \oplus N_R \oplus N_R' \oplus K_{old}) \oplus K_{old} = PRNG(EPC_S \oplus N_R \oplus D \oplus K_{old}) \oplus K_{old} = M1 = M1'$.
   - Verifies whether $N_T' \oplus PRNG(C_{old}' \oplus K_{old}') = E'$, where:
   - $N_T' = D' \oplus K_{old} = N_T \oplus N_R \oplus N_R' \Rightarrow N_T' \oplus PRNG(C_{old} \oplus K_{old}) = N_T \oplus N_R \oplus N_R' \oplus PRNG(C_{old} \oplus K_{old}) = E'$.
   - Authenticates the adversary as a legitimate tag and computes $M2'$ and Info as follows, and forwards them to the reader:
     $M2' \leftarrow PRNG(EPC_S \oplus N_T') \oplus P_{old}'$ and Info ← DATA ⊕ RID
   - Since X=old, updates the back-end database as follows:
     $C_{new}' \leftarrow PRNG(N_T' \oplus N_R')$.
5. Once the reader receives the message, it extracts DATA and forwards M2 to the expected tag, which is the adversary.

Following the given attack, the adversary is authenticated by the back-end database as a legitimate tag with a probability of 1, while the complexity of the attack is only two protocol runs with negligible time and memory requirements [7].

3.3. Data Desynchronization Attack

In 2013 Deng and Zhu [11] have shown that the Habibi et al.'s protocol, can't resist the data desynchronization attack either. Before the implementation of the data desynchronization attack, Adversary $\mathcal{A}$ needs to carry out a secret information disclosure attack that has been described in section 3.2. Thus $\mathcal{A}$ can disclose all the Secret Information of T, including EPCs, Ki and Pi. Then $\mathcal{A}$ can easily launch the data desynchronization attack. The process of the data desynchronization attack is shown as follows. Firstly, $\mathcal{A}$ launches the secret information disclosure attack and retrieves any secret information in T, including EPC_S, K_i and P_i. Secondly, $\mathcal{A}$ eavesdrops the random number N_R generated by R and values C_i, M1, D, E generated by T in the following protocol run, and it intercepts the message C_i, M1, D, E from the tag to the reader. Thirdly, $\mathcal{A}$ Computes $N_T = D \oplus K_i$, $M2 = PRNG(EPC_S \oplus N_T) \oplus P_i$ and forwards M2 to T. Once T receives M2, it authenticates Server S and updates the contents kept inside as $K_{i+1} \leftarrow PRNG(K_i)$, $P_{i+1} \leftarrow PRNG(P_i)$, $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$. Therefore, the tag has refreshed the secrets K_i, P_i, C_i while the back-end server will not do it. Thus, the shared secret between the tag and the back-end server may not be the same, which can bring system to a mess. After a successful data desynchronization attack, because $\mathcal{A}$ makes S and the valid tag T share the different secrets, S will not be authorized by T and T will not be authorized by S yet [11].

3.4. Traceability Attack

Castro et al. [7] have shown that the Habibi et al.'s protocol, like the original protocol, puts at risk the location privacy of tags' holders because it is possible to track tags with a probability of 1 – between two successful runs of the authentication protocol. The following properties of the protocol are enough to trace a given tag T_i, as long as it has not updated its internal values:

1. When the reader or possibly the adversary $\mathcal{A}$, which supplants a legal reader in a mutual authentication session, sends a random number N_R to the tag, it will answer with M1, C_i, where C_i is the tag's index in the back-end database and will remain fixed as long as the tag does not participate in another successful protocol run toupdate its internal values.
2. Given that the tag's reply to the reader's (or adversary) query includes D and E,

Where $D = N_T \oplus K_i$ and $E = N_T \oplus PRNG(C_i \oplus K_i)$. It can be seen that if $\mathcal{A}$ computes Y as follows:
$Y \leftarrow D \oplus E = N_T \oplus K_i \oplus N_T \oplus PRNG(C_i \oplus K_i) = K_i \oplus PRNG(C_i \oplus K_i)$ then Y only depends on K_i and C_i and these ones will remain fixed as long as the tag does not execute a new updating phase. Hence, Y can be used as a value to perfectly trace T_i [7].

4. Enhanced protocol

In order to eliminate the mentioned vulnerabilities in 3.1, 3.2 and 3.3 sections, we can modify the message E as: $E = N_T \oplus PRNG(C_i \oplus K_i) \oplus P_i$. Although the cited vulnerabilities are fixed by the above modification, but the traceability problem that has been discussed in section 3.4, still will be unsolved. Hence, we need to reconstruct the message E as following: $E=PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i$ to provide a secure protocol against all cited attacks.

Fig.1, illustrates the (i+1)th authentication phase of proposed protocol. The detailed steps of the authentication phase are presented as follows.
1) The reader R generates a random number N_R and sends it to the tag T.
2) T receives N_R, generates a random number N_T, computes M1, D, E and finally sends
M1, D, E and C_i to R, where $M1 = PRNG(EPC_S \oplus N_R \oplus N_T) \oplus K_i$ and $D = N_T \oplus K_i$ and $E = PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i$.
3) When R receives the message, it computes $V= h(RID \oplus N_R)$ and forwards M1, D, C_i, E, N_R, V to the back-end server S.
4) After S receiving M1, D, C_i, E, N_R, and V, it proceeds as follows.
- For each RID stored in the database, it computes $h(RID \oplus N_R)$ and compares it with the received V to verifies R legitimacy.
- If $C_i = 0$, which means that it is the first access to the tag, it proceeds as follows, iteratively:
  (a) Picks up an entry (K_{old} , P_{old} , C_{old} , K_{new} , P_{new} , C_{new} , RID, EPCs, DATA) stored in database.
  (b)Verifies whether $M1 \oplus K_{old} = PRNG(EPC_S \oplus N_R \oplus D \oplus K_{old})$ or $M1 \oplus K_{new} = PRNG(EPC_S \oplus N_R \oplus D \oplus K_{new})$, and marks X as old or new provided that the verification process is satisfied based on the new record or the old record.
- Otherwise, S uses C_i as an index to find the corresponding record in the database and verify whether $PRNG(EPC_S \oplus N_R \oplus D \oplus K_X) \oplus K_X = M1$. If "No" the protocol aborts.
- Verify whether $PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i = E$. If "No" the protocol aborts.
- Computes M2 and Info and forwards them to R, where $M2 = PRNG(EPC_S \oplus N_T) \oplus P_X$

ACSIJ Advances in Computer Science: an International Journal, Vol. 3, Issue 2, No.8 , March 2014
ISSN : 2322-5157
www.ACSIJ.org

and Info = DATA ⊕ RID.

- If X=new, updates the database as follows:
  $K_{old} \leftarrow K_{new}$, $K_{new} \leftarrow PRNG(K_{new})$,
  $P_{old} \leftarrow P_{new}$, $P_{new} \leftarrow PRNG(P_{new})$,
  $C_{old} \leftarrow C_{new}$, $C_{new} \leftarrow PRNG(N_T \oplus N_R)$.
- Else
  $C_{new} \leftarrow PRNG(N_T \oplus N_R)$.

5) Once R receives the message, it extracts DATA as $Info \oplus RID$ and forwards M2 to T.

6) When T receives the message, it verifies whether $PRNG(EPCs \oplus N_T) = M2 \oplus P_i$.

If "No" the protocol aborts. Else T authenticates S and updates the contents kept inside as $K_{i+1} \leftarrow PRNG(K_i)$, $P_{i+1} \leftarrow PRNG(P_i)$, $C_{i+1} \leftarrow PRNG(N_T \oplus N_R)$.

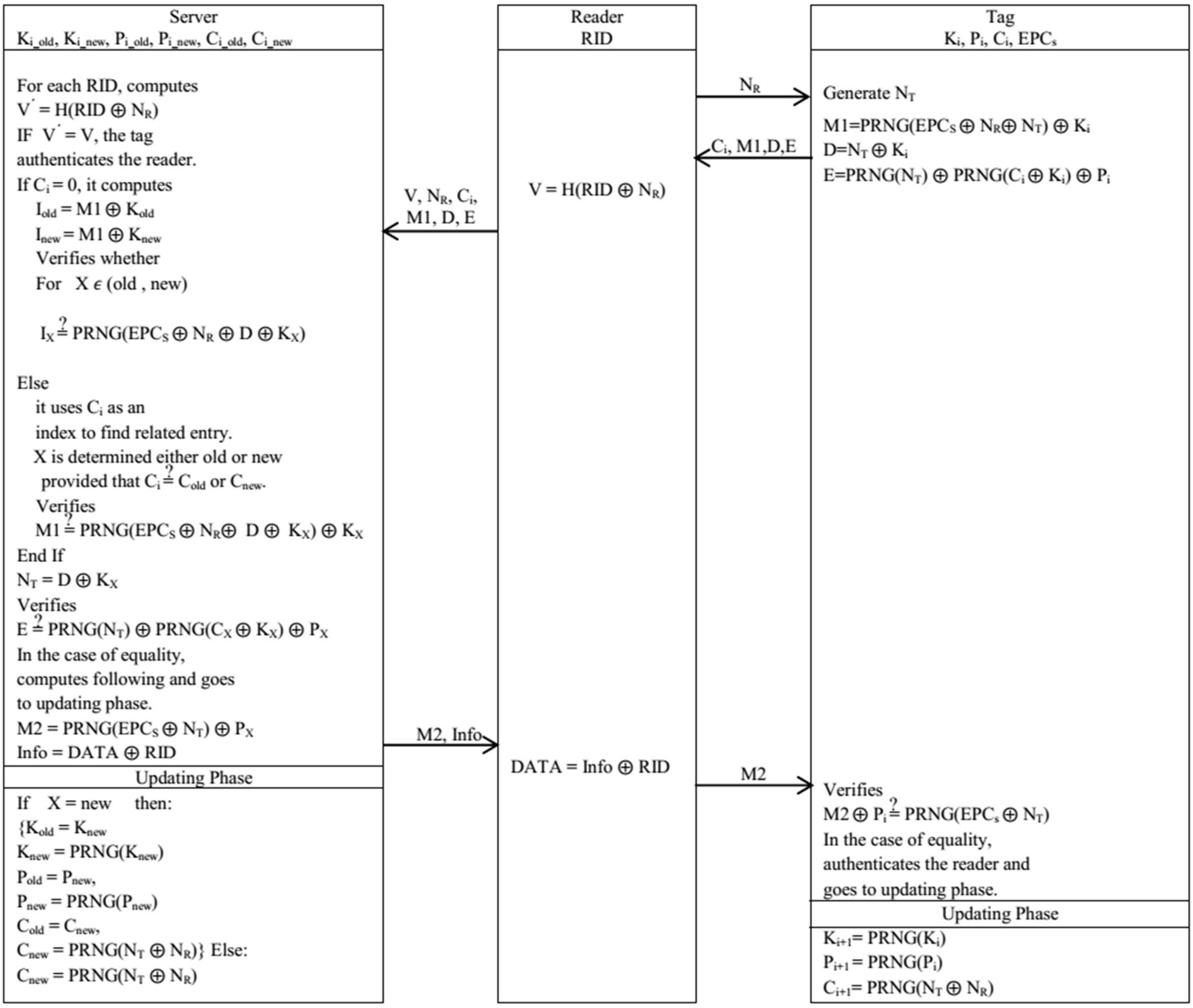

Fig. 1 (i +1)th authentication phase of proposed protocol

4.1. Security analysis of enhanced protocol

In this section security and privacy of proposed protocol is evaluated against various threats.

4.1.1. Secret Information Disclosure Attack

The proposed protocol resists to this attack, because of XOR P_i with E. By this modification, step 2 of this attack has not established, because the adversary does not know the value of P_i and cannot obtain N_T and K_X values as fallow:

$\forall$ i=0...N_d does as follows:
- $K_i \leftarrow i$,
- $N_T \leftarrow D \oplus K_i$,
- $E \neq N_T \oplus PRNG(C_i \oplus K_i) \oplus P_i$.

4.1.3. Replay Attack

Updating the secret values in each authentication process and the prevention of Secret Information disclosure, and in particular using random values N_R and N_T for making the transition massages, an adversary cannot send obtained information in the next round of the authentication instead of legal tag, because of variation of massages.

4.1.4. Traceability Attack

In the proposed protocol to resist this attack that has been discussed in section3.4, N_T has been replaced by $PRNG(N_T)$ in the message E, so in the proposed attack the result of XOR messages D and E is not fixed because of the random value (N_T) has not been deleted.
$D = N_T \oplus K_i$
$E = PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i$
$Y = D \oplus E = N_T \oplus K_i \oplus PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i$

As a result the value of Y is not fixed in each authentication phase although the updating phase has not been executed.

4.1.5. Privacy

In proposed protocol the privacy problem has been solved because of avoidance of Secret Information disclosure and traceability attacks.

4.1.6. DoS Attack

If an adversary prevents the tag from updating it's secret information by intercepting M2, the server is asynchronous with the tag and at a result the communication between them will be intercepted, but In this case by keeping C_{old} value in the database, in the next authentication session the server supposed that tag authentication process in the previous session is not completed successfully. Then it authenticates the tag by it's C_{old} and only updates it's C_{new}.

4.1.7. Tag Impersonation Attack

4.1.2. Desynchronization Attack

An adversary requires tag's Secret Information for desynchronization attack that has been avoided in proposed protocol. Also keeping K_{old}, C_{old}, P_{old} values in back-end server can help to avoid desynchronization Attack occurrence.

The proposed protocol resists to tag impersonation attack that has been discussed in section 3.2, by changing the structure of E as fallowing: $E=PRNG(N_T) \oplus PRNG(C_i \oplus K_i) \oplus P_i$.
By this modification the back-end server cannot confirm E' as below:
$PRNG(N_T') \oplus PRNG(C_{old} \oplus K_{old}) \oplus P_i = E'$
$N_T' = D' \oplus K_{old} = N_T \oplus N_R \oplus N_R' \Rightarrow PRNG(N_T') \oplus PRNG(C_{old} \oplus K_{old}) \oplus P_i = PRNG(N_T \oplus N_R \oplus N_R') \oplus PRNG(C_{old} \oplus K_{old}) \oplus P_i \neq E'$.

4.1.8. Database Loading

In this protocol, similar to previous version, C_i is used as an index to access the database which requires record-by-record operations and verifications only in the first access and the index for the tag can be set accordingly. As for any later on accesses, only C_i will be needed as an index. Thus, the performance of the system has not been changed.

Table1. Comparison of authentication protocols

	Chien and Chen	*Yeh et al*	*Habibi et al*	*Proposed protocol*
Desynchronization Attack	Insecure	Insecure	Insecure	Secure
Replay Attack	secure	Secure	Secure	Secure
Tracking Attack	Insecure	Insecure	Insecure	Secure
Privacy	No provide	No provide	No provide	Secure
DoS Attack	Insecure	Insecure	Insecure	Secure
Tag Impersonation Attack	Insecure	Insecure	Insecure	Secure
Database Loading	High	Low	Low	Low

5. Conclusions

In this paper, we demonstrated some security problems of Habibi et al.'s RFID authentication protocol. We discussed a powerful and practical attack on this protocol which is secret information disclosure. This attack leads to desynchronization attack. Moreover, we explained the tag impersonation and traceability attacks on this protocol. To eliminate all cited vulnerabilities, we enhanced this protocol by reconstructing the message E in a new way. Finally the enhanced protocol, has been compared with the existing EPC-C1-GEN2-based RFID authentication protocols in terms of security and privacy.

The comparison results showed that the enhanced protocol can enhance the security and privacy in RFID systems.

Reference

[1] D. Henrici, "RFID security and privacy: concepts, protocols, and architectures". 1st ed. New York, Springer, 2008.
[2] Class-1 generation 2 UHF air interface protocol standard version 1.2.0, Gen2, 2008.
[3] EPC Tag data standard version 1.4, 2008, http://www.epcglobalinc.org/standards/. "Yearly report on algorithms and keysizes, Technical Report D.SPA.13Rev.1.0, ICT-2007-216676, In Gen2, ECRYPT", 2010.
[4] Information technology Radio frequency identification for item management, Part 6, "parameters for air interface communications at 860 MHz to 960MHz". http://www.iso.org. 2005.
[5] D.V. Bailey, and A. Juels, "Shoehorning security into the EPC tag standard", In R. D. Prisco, and M. Yung, editors, SCN, of Lecture Notes in Computer Science, Springer, 2006, Vol. 4116, pp. 303–320.
[6] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-T apiador, and A. Ribagorda, "RFID specification revisited", In The internet of things:From RFID to The Next-Generation Per vasive Networked Systems, Taylor & Francis Group, 2008, pp.311–346.
[7] C.H.C. Julio, P.L. Pedro, S. Masoumeh, B. Nasour, and N. Majid,"Another Fallen Hash-Based RFID Authentication Protocol", Proceedings of WISTP, Vol.7322, 2012, pp. 29-37.
[8] C.L. Chen, and Y.Y. Deng, "Conformation of EPC class 1 and generation 2 standards RFID system with mutual authentication and privacy protection", Engineering Applications of Artificial Intelligence,Vol.22, No. 8, 2009, pp.1284–1291.
http://www.epcglobalinc.org/standards/.
[9] G.Tsudik,"YA-TRAP: yet another trivial RFID authentication protocol", 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW' 06), 2006, pp. 640-643.
[10] H.Y. Chien, and C.H. Chen, "Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards", Computer Standards and Interfaces, Vol. 29, No. 2, 2007, pp. 254–259.
[11] M. Deng, W. Zhu, "Desynchronization Attacks on RFID Security Protocols", in: TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 11, No. 2, 2013, pp. 681-688.
[12] M. H. Habibi, M. R. Alaghband, and M. R. Aref, "Attacks on a lightweight mutual authentication protocol under EPC C-1 G-2 standard", In C. A. Ardagna, and J. Zhou, editors, WISTP, Springer, 2011, vol. 6633 of Lecture Notes in Computer Science, pp. 254–263.
[13] S. Karthikeyan, and M. Nesterenko, "RFID security with out extensive crypto-graphy", In: Proceedings of the Third, ACM Workshop on Security of Ad Hoc and Sensor Networks, 2005,63–67.
[14] T. C. Yeh, Y. J. Wang, T. C. Kuo, and S. S. Wang, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard", Expert Systems with Applications, Vol. 37, 2010, pp. 7678-7683.