Another Fallen Hash-Based RFID Authentication Protocol

Julio Cesar Hernandez-Castro[1], Pedro Peris-Lopez[2], Masoumeh Safkhani[3], Nasour Bagheri[4], and Majid Naderi[3]

[1] School of Computing, Portsmouth University, UK
Julio.Hernandez-Castro@port.ac.uk
[2] Computer Science Department, Carlos III University of Madrid, Spain
pperis@inf.uc3m.es
[3] Electrical Eng. Department, Iran University of Science and Technology, Tehran, Iran
{M_Safkhani,M_Naderi}@iust.ac.ir
[4] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran
NBagheri@srttu.edu

Abstract. In this paper, we scrutinize the security of an RFID protocol [9], which has been recently proposed, and show important vulnerabilities. Our first attack is a passive one that can disclose all secret information stored on the tags' memory. We only need to eavesdrop one session of the protocol between a tag and a legitimate reader (connected to the back-end database) and perform $O(2^{17})$ off-line evaluations of the *PRNG*-function – while the authors wrongly claimed the complexity of any such attack would be around 2^{48} operations. Although the extracted information is enough to launch other relevant attacks and thus to completely rule out any of the protocol's security claims, we additionally present several attacks using alternative strategies that show the protocol is flawed in more than one way and has many exploitable weaknesses. More precisely, we present a tag impersonation attack that requires the execution of only two runs of the protocol, and has a success probability of 1. It must be noted that this attack is, however, not applicable to the original protocol that the authors attempted to improve so, in a way, their improvement is not such. Finally, we show two approaches to trace a tag, as long as it has not updated its secret values. For all the above, we conclude that the improved protocol is even less secure than the original proposal, which is also quite insecure, and cannot be recommended.

Keywords: RFID, EPC-C1G2, Authentication, Secret Disclosure, Impersonation, Traceability.

1 Introduction

Radio Frequency Identification (RFID) is a wireless technology which can be employed to identify or track objects in various applications. Some common applications are animal tracking, retail, supply chain management in wholesale stores, library access control, toll payments, theft prevention, human implants, and e-passports. A typical RFID system includes a reader and a number of tags, which may range from the high

I. Askoxylakis, H.C. Pöhls, and J. Posegga (Eds.): WISTP 2012, LNCS 7322, pp. 29–37, 2012.
© IFIP International Federation for Information Processing 2012

end battery-powered ones with Wi-Fi capabilities, to the low-cost that are quite constrained in resources and have no internal power, harvesting it from the readers. The tag generally includes some information related to the tag holder, and can be read/modified by the reader, which is normally securely connected to a back-end database through classical means (e.g. SSL). This technology is expected to replace barcodes in grocery and retail stores in the near future.

However, despite the multiple benefits mentioned above, security and privacy are the main concerns that slow down the rapid and widespread deployment of this technology. For instance, regarding these security concerns, only the authorized readers should be able to read or modify the information stored on the tags, only valid tags should be authenticated by a legitimate reader and it should be infeasible for a fake tag to impersonate a legitimate one. To address these multiple security and privacy requirements, several RFID mutual authentication protocols and their security analysis have already been proposed in literature, e.g. [7, 10, 11, 14]. In addition, there are several interconnected standards for RFID systems, and among them EPC global and ISO have played a major role. The Electronic Product Code Class-1 Generation-2 specification [6, 8] (EPC-C1G2 in short) was announced in 2004 by EPC Global and ratified by ISO [12]. However, later security analysis carried out on the EPC-C1 G2 specification demonstrated several security concerns [1, 13]. Researchers, motivated by this, have proposed many EPC-compliant schemes –in an attempt to correct the weaknesses of the standard and improve its security– and have analyzed the security of these new schemes [2–5, 9, 15]. Among them, one of the most recent proposals is a protocol proposed by Habibi *et al.* [9], which is an improvement to the Yeh *et al.* 's protocol [15]. Specifically, the authors analyzed the security of Yeh *et al.* 's protocol and proposed an improved version as a repair for the attacks they found. This new proposal is the main concern of this paper.

In this paper, we show that Habibi *et al.* did not succeed in their attempt, and the proposed protocol is at least as insecure as its predecessor. More precisely, they decreased the security margin of the original protocol rather than improve it, because it is possible to apply an efficient tag impersonation on the revised protocol which is not applicable to the original protocol. In addition to that, all the security problems of the original protocol remain unsolved.

Paper Organization: In § 2 some preliminaries and notations are introduced. We describe the improved Yeh *et al.* 's protocol proposed by Habibi *et al.* in § 3. A secret information disclosure attack is presented in § 4. § 5 and § 6 describe tag impersonation and traceability attacks, respectively. Finally, in § 7 we extract some interestings conclusions.

2 Preliminaries

Throughout the paper, we use the following notation:

- EPC_s: The 96 bits of EPC code are divided into six 16-bit blocks, and then these six blocks are XORed to form EPC_s.
- $DATA$: The corresponding information for the tag, kept in the back-end database.

- K_i: The 16-bit authentication key stored in the tag to be authenticated by the back-end database at the $(i+1)^{th}$ phase of authentication.
- P_i: The 16-bit access key stored in the tag to authenticate the back-end database at the $(i+1)^{th}$ phase of authentication.
- K_{old} and K_{new}: The old and new authentication keys, respectively, stored in the back-end database.
- P_{old} and P_{new}: The old and new access keys, respectively, stored in the back-end database.
- C_i: The 16-bit index of the record of the ith tag's information in the back-end database, stored in the tag.
- C_{old} and C_{new}: The old and new back-end database indexes for the ith tag, respectively, stored in the back-end database .
- X: The value kept as either *new* or *old* to show which key in the record of the back-end database is matched with the ones on the tag.
- $B \longleftarrow A$: Assign the value of A to B.
- N_T and N_R: 16-bit random numbers (nonces) that are generated by the tag and the reader, respectively.
- $\oplus$: Exclusive-OR operation.
- RID: The reader identification number.
- $PRNG$: a 16-bit pseudo-random number generator.
- $H(.)$: A secure cryptographic hash function.

3 Protocol Description

In this section we give a brief description of Habibi *et al.* 's protocol – see the original paper [9] for further details. This protocol has two phases: an initialization phase and an $(i+1)^{th}$ authentication phase, which are described as follows:

Initialization Phase: In this phase, the manufacturer generates random values for K_0, P_0 and C_0 respectively and sets the values of the record in the tag, i.e., $K_i = K_0$, $P_i = P_0, C_i = C_0$ and the corresponding record in the back-end database $K_{old} = K_{new} = K_0, P_{old} = P_{new} = P_0, C_{old} = C_{new} = 0$.

Authentication Phase: The authentication phase of Habibi *et al.* 's protocol, in its $(i+1)^{th}$ run, depicted in Fig. 1 in Appendix, is as follow:

1. The reader generates a random number N_R and sends it to the tag.
2. The tag receives N_R, generates a random number N_T, computes M_1, D, E as shown below and finally sends M_1, D, C_i and E to the reader:
   $M_1 \longleftarrow PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$ and $D \longleftarrow N_T \oplus K_i$ and
   $E \longleftarrow N_T \oplus PRNG(C_i \oplus K_i)$.
3. Once the reader receives the message, it computes $V = H(RID \oplus N_R)$ and forwards M_1, D, C_i, E, N_R, V to the back-end database.
4. The back-end database receives M_1, D, C_i, E, N_R and V. After receiving these values, it proceeds as follows:
   - For each RID stored in the database (DB), it computes $H(RID \oplus N_R)$ and compares it with the received V to verifies the reader legitimacy.

- If $C_i = 0$, which means that it is the first access to the tag, it proceeds as follows, iteratively:
  - Picks up an entry (K_{old}, P_{old}, C_{old}, K_{new}, P_{new}, C_{new}, RID, EPS_s, $DATA$) stored in database.
  - Verifies whether $M_1 \oplus K_{old} \stackrel{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{old})$ or $M_1 \oplus K_{new} \stackrel{?}{=} PRNG(EPC_s \oplus N_R \oplus D \oplus K_{new})$. If "Yes" marks X as *old* or *new* provided that the verification process is satisfied based on the new record or the old record.
- Otherwise, it uses C_i as an index to find the corresponding record in the database and verify whether $PRNG(EPC_s \oplus N_R \oplus D \oplus K_X) \oplus K_X \stackrel{?}{=} M_1$. If "No" the protocol aborts.
- Verify whether $N_T \oplus PRNG(C_X \oplus K_X) \stackrel{?}{=} E$. If "No" the protocol aborts.
- Computes M_2 and $Info$ as follows and forwards them to the reader: $M_2 \longleftarrow PRNG(EPC_s \oplus N_T) \oplus P_X$ and $Info \longleftarrow DATA \oplus RID$
- If $X = new$, updates the database as follows: $K_{old} \longleftarrow K_{new}$, $K_{new} \longleftarrow PRNG(K_{new})$, $P_{old} \longleftarrow P_{new}$, $P_{new} \longleftarrow PRNG(P_{new})$, $C_{old} \longleftarrow C_{new}$, $C_{new} \longleftarrow PRNG(N_T \oplus N_R)$.
- Else, $C_{new} \longleftarrow PRNG(N_T \oplus N_R)$.

5. Once the reader receives the message, it extracts $DATA$ as $Info \oplus RID$ and forwards M_2 to the tag.
6. Once the tag receives the message, it proceeds as follows:
   - Verifies whether $PRNG(EPC_s \oplus N_T) \stackrel{?}{=} M_2 \oplus P_i$. If "No" the protocol aborts.
   - Authenticates the back-end database.
   - Updates the contents kept inside as $K_{i+1} \longleftarrow PRNG(K_i)$, $P_{i+1} \longleftarrow PRNG(P_i)$ and $C_{i+1} \longleftarrow PRNG(N_T \oplus N_R)$.

It must be noted that the only difference between the above protocol and the original protocol, proposed by Yeh *et al.* [15], is that in the original protocol M_1 is computed as $M_1 = PRNG(EPC_s \oplus N_R) \oplus K_i$.

4 Secret Information Disclosure Attack

In this section we present an efficient and passive attack that retrieves any secret information in the tag, including EPC_s, K_i and P_i. The main observation, which is the milestone of the given attack, is the fact that given $Y = PRNG(X)$ and the assumptions that the $PRNG$-function is a public function, and the length of Y and X is 16-bit, then it is possible to do an exhaustive search and find X as a pre-image of Y in the cost of at most 2^{16} off-line evaluations of $PRNG$. Following this observation, and given the fact that the tag T_i communicates with a legitimate reader R_i, an adversary ($\mathcal{A}$) can disclose all the secret parameters of T_i as follows:

1. Eavesdrops one session of the protocol and stores all the exchanged messages: $N_R, C_i, M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i, D = N_T \oplus K_i, E = N_T \oplus PRNG(C_i \oplus K_i)$ and $M_2 = PRNG(EPC_s \oplus N_T) \oplus P_X$.

2. $\forall\, i = 0, \ \ldots, \ 2^{16} - 1$ does as follows:
   - $K_i \longleftarrow i$ and $N_T \longleftarrow D \oplus K_i$,
   - If $E = N_T \oplus PRNG(C_i \oplus K_i)$ then return K_i and N_T.
3. For the returned values of K_i and N_T from Step 2 and $\forall\, i = 0, \ \ldots, \ 2^{16} - 1$ does as follows:
   - $EPC_s \longleftarrow i$,
   - If $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_i$ then return EPC_s.
4. For the returned values of K_i and N_T from Step 2 and EPC_s from Step 3 assigns $M_2 \oplus PRNG(EPC_s \oplus N_T)$ to P_i and returns the following values:
   $P_{old} = P_i$, $P_{new} = PRNG(P_i)$,$K_{old} = K_i$, $K_{new} = PRNG(K_i)$, $C_{old} = C_i$.

The complexity of the given attack is limited to eavesdropping one session of the protocol between a tag and a legitimate reader, and perform 2^{17} evaluations of the *PRNG*-function. However, the adversary succeeds in its attack if it comes up with only one pre-image in each of Steps 2 and 3 of the given attack (it must be noted that the existence of at least one pre-image in each step is guaranteed). Otherwise, it should repeat the attack several times to come up with an unique solution. To increase the efficiency of the proposed attack, the adversary can block M_2 in the last Step of the protocol to avoid the updating of the secret values. In this case two runs of the protocol should be fairly enough to extract all given parameters.

Given all secret values of the tag, it would be easy to launch other relevant attacks with a success probability of 1, and the cost of one execution of the protocol (e.g. traceability, tag impersonation, reader impersonation and de-synchronization).

Remark 1. It must be noted that a similar attack was applied by Habibi *et al.* [9] on the original protocol of Yeh *et al.* and the improved protocol was proposed to overcome this weaknesses. In their security analysis the authors claimed that the complexity of disclosing the secret information in their improved protocol is 2^{48} evaluations of the *PRNG* function. Nevertheless, we present an efficient attack which retrieves all secret parameters with a cost of 2^{17} evaluations, which explicitly contradicts their claims.

Although the above attack ruins all the security properties objectives of the protocol, we continue presenting other attacks based on different strategies.

5 Tag Impersonation Attack

Tag impersonation attack is a forgery attack that leads to the identification of spoofed tags by a legitimate reader. In this section we show 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 do as follows:

Phase 1 (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, M_1, 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, EPS_s, $DATA$) and the tag record includes (K_{new}, P_{new}, C_{new}, EPS_s), where: $K_{new} = PRNG(K_{old})$, $P_{new} = PRNG(P_{old})$, $C_{new} = PRNG(N_T \oplus N_R)$, $M_1 = PRNG(EPC_s \oplus N_R \oplus N_T) \oplus K_{old}$, $D = N_T \oplus K_{old}$ 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 M'_1, D', C'_i and E' where: $M'_1 = M_1 = PRNG(EPC_s \oplus N'_R \oplus N_T) \oplus K_{old}$, $C'_i = C_{old}$, $D' = D \oplus N_R \oplus N'_R = N_T \oplus K_i \oplus N_R \oplus N'_R$ and $E' = E \oplus N_R \oplus N'_R = N_T \oplus PRNG(C_{old} \oplus K_{old}) \oplus N_R \oplus N'_R$.
3. Once the reader receives the message, it computes $V = H(RID \oplus N'_R)$ and forwards M'_1, 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.
   - We 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 $PRNG(EPC_s \oplus N'_R \oplus D' \oplus K'_{old}) \oplus K_{old} \stackrel{?}{=} M'_1$, where: $PRNG(EPC_s \oplus N'_R \oplus D' \oplus K_{old}) \oplus K_{old} =$ $PRNG(EPC_s \oplus N'_R \oplus D \oplus N_R \oplus N'_R \oplus K_{old}) \oplus K_{old} =$ $PRNG(EPC_s \oplus N_R \oplus D \oplus K_{old}) \oplus K_{old} = M_1 = M'_1$.
   - Verifies whether $N'_T \oplus PRNG(C'_{old} \oplus K'_{old}) \stackrel{?}{=} E'$, 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 M'_2 and $Info$ as follows, and forwards them to the reader: $M'_2 \longleftarrow PRNG(EPC_s \oplus N'_T) \oplus P'_{old}$ and $Info \longleftarrow DATA \oplus RID$
   - Since $X = old$, updates the back-end database as follows: $C'_{new} \longleftarrow PRNG(N'_T \oplus N'_R)$.
5. Once the reader receives the message, it extracts $DATA$ and forwards M_2 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. It is worth to note that the given attack is not applicable to the original protocol of Yeh *et al.* and the complexity of the best known tag impersonation attack against the original protocol is 2^{16} evaluations of *PRNG* function [9]. It shows that Habibi *et al.* have decreased the security of the original protocol while trying to improve it – at least from this attack's point of view.

6 Traceability Attack

In this section, we show that the improved Yeh *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 M_1, 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 to update 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:
   $Y \longleftarrow D \oplus E = N_T \oplus K_i \oplus N_T \oplus PRNG(C_i \oplus K_i) = K_i \oplus PRNG(C_i \oplus K_i)$
   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.

It must be noted that this attack also works against the original protocol of Yeh *et al.*

7 Conclusions

In this paper we analyzed the security of the improved Yeh *et al.* 's protocol, designed to be compliant with the EPC-C1G2 standard, and being one of the most recent proposed protocols in this area. Our main attack is a passive full disclosure attack which can retrieve efficiently all the secret parameters of the tag. The cost of this attack is the eavesdropping of one protocol session and the performing of $O(2^{17})$ off-line evaluations of the *PRNG*-function – while Habibi *et al.* claimed $O(2^{48})$ evaluations are needed for any such attack. This attack is so powerful that it ruins all the security properties claimed by the proposed scheme. To complete this analysis, and following different strategies, we also present tag impersonation and traceability attacks that prove that these protocols are flawed in more than one way and probably do not admit an easy fixing. Summarizing, in this paper we show how the improved protocol proposed by Habibi *et al.* is more insecure that the one they tried to correct, which is regrettably a too common occurrence in the area.

References

1. Bailey, D.V., Juels, A.: Shoehorning Security into the EPC Tag Standard. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 303–320. Springer, Heidelberg (2006)
2. Burmester, M., de Medeiros, B.: The Security of EPC Gen2 Compliant RFID Protocols. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 490–506. Springer, Heidelberg (2008)

3. Burmester, M., de Medeiros, B., Munilla, J., Peinado, A.: Secure EPC Gen2 Compliant Radio Frequency Identification. In: Ruiz, P.M., Garcia-Luna-Aceves, J.J. (eds.) ADHOC-NOW 2009. LNCS, vol. 5793, pp. 227–240. Springer, Heidelberg (2009)
4. Chen, C.-L., Deng, Y.-Y.: Conformation of EPC Class-1 Generation-2 standards RFID system with mutual authentication and privacy protection. Eng. Appl. of AI 22(8), 1284–1291 (2009)
5. Chien, H.-Y., Chen, C.-H.: Mutual authentication protocol for RFID conforming to EPC Class-1 Generation-2 standards
6. Class-1 Generation-2 UHF air interface protocol standard version 1.2.0, EPCGlobal (2008), `http://www.epcglobalinc.org/standards/`
7. Duc, D.N., Kim, K.: Defending RFID authentication protocols against DoS attacks. Computer Communications 34(3), 384–390 (2011)
8. EPC Tag data standard version 1.6, EPCGlobal (2011), `http://www.epcglobalinc.org/standards/`
9. Habibi, M.H., Alagheband, M.R., Aref, M.R.: Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard. In: Ardagna, C.A., Zhou, J. (eds.) WISTP 2011. LNCS, vol. 6633, pp. 254–263. Springer, Heidelberg (2011)
10. Hung-Yu, C.: SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. IEEE Transactions on Dependable and Secure Computing 4(4), 337–340 (2007)
11. Chien, H.Y.: Secure access control schemes for RFID systems with anonymity. In: Proceedings of MDM, p. 96 (2006)
12. Information technology Radio frequency identification for item management. Part 6: parameters for air interface communications at 860 MHz to 960MHz- (2005), `http://www.iso.org`
13. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: RFID specification revisited. In: The Internet of Things: From RFID to The Next-Generation Pervasive Networked Systems, pp. 6:311–6:346. Taylor & Francis Group (2008)
14. Weis, R.-D.E.S., Sarma, S.: Security and privacy aspects of low-cost radio frequency identification systems. In: Proceedings of WiCom, pp. 2078–2080 (2007)
15. Yeh, T.-C., Wang, Y.-J., Kuo, T.-C., Wang, S.-S.: Securing RFID systems conforming to EPC Class-1 Generation-2 standard. Expert Syst. Appl. 37(12), 7678–7683 (2010)

Appendix

A Habibi *et al.*'s Protocol Description

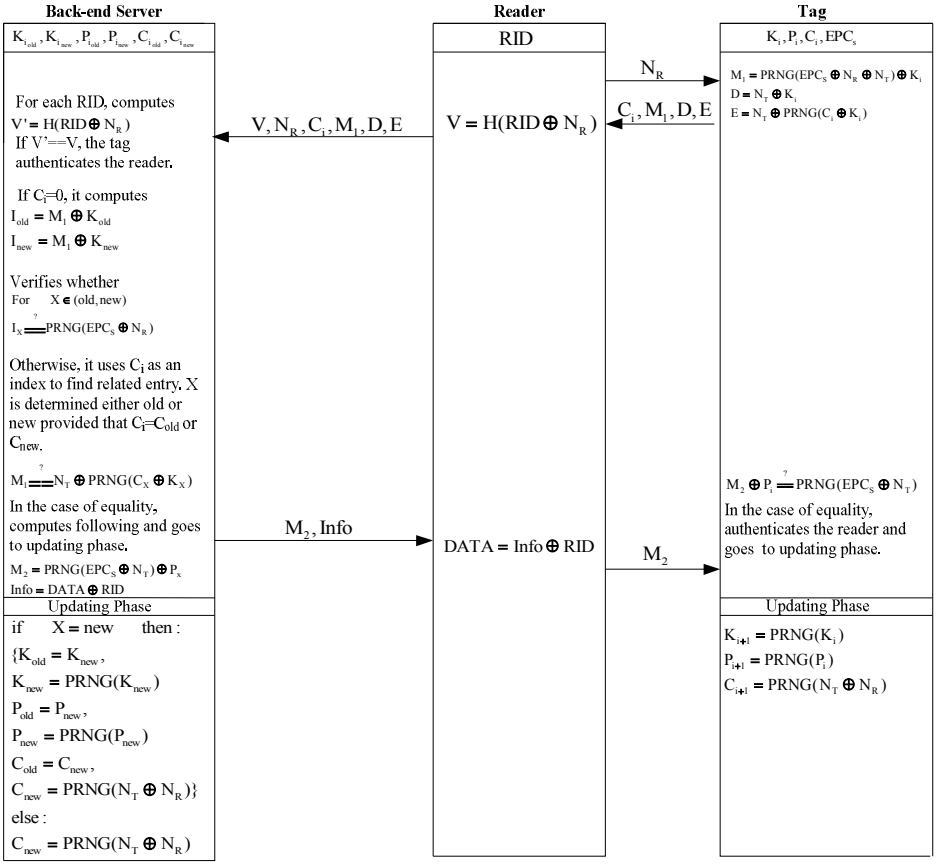

Fig. 1. Improvement of Yeh *et al.*'s Authentication Protocol by Habibi *et al.* [9]