Vulnerability Analysis of a Mutual Authentication Protocol Conforming to EPC Class-1 Generation-2 Standard

Mu'awya Naser
Khalifa City Women College
Higher College of Technology HCT
Abu Dhabi, UAE
muawya.aldalaien@hct.ac.ae

Rahmat Budiarto
College of Comp. Science and Information Technology
Albaha University
Albaha, Kingdom of Saudi Arabia
rahmat@bu.edu.sa

Ismat Aldmour
College of Comp. Science and Information Technology,
Albaha University
Albaha, Kingdom of Saudi Arabia
iaaldmour@bu.edu.sa

Pedro Peris-Lopez
Computer Security Lab (COSEC), Computer Science Department
Carlos III University of Madrid
Madrid, Spain
pperis@inf.uc3m.es

***Abstract*—In this paper we scrutinize the security properties of an RFID authentication protocol conforming to the EPC Class-1 Generation-2 standard. The protocol is suitable for Gen-2 passive tags and requires simple computations. The authors claim that the scheme provides privacy protection and authentication and offers resistant against commonly assumed attacks. We propose a de-synchronization and an impersonation attack in which the disclosing of the secret information (i.e. secret key and static identifier) shared between the tag and the reader is unnecessary to success in these attacks.**

***Keywords*—**

1. INTRODUCTION

Radio Frequency IDentification (RFID) is a technology highly demanded in numerous applications and domains and therefore is under a continuous and rapid development [1-4]. Securing RFID tags against security threats is considered the main obstacle facing the widespread adoption of RFID technology [5-8], where hundreds of RFID protocols have been proposed and focused on providing a secure contact between readers and tags over the insecure radio channel. Nevertheless, due to the limitations of tags in terms of circuitry (gate equivalents), storage and power consumption, the design of an efficient and secure mutual authentication protocol presents an immense challenge. It is even more challenging for low-cost technologies such as lightweight RFID security protocols in which tags possesses stronger limitations. Among the set of risks linked to RFID technology, privacy and de-synchronization are the most challenging as the majority of designed protocols fail to offer protection against these two threats.

Yeh et al.'s protocol [9] aims to secure EPC Class-1 Generation-2 standard. Similar to many previously proposed protocols, it can be categorized under the class of lightweight mutual authentication protocol following the classification proposed in [10]. In this category, it is assumed that tags can generate a random number but they do not have the computational resources to support on-board hash function. On the other hand and similar to other lightweight RFID authentication protocols, Yeh et al.'s scheme is designed with a new parameter representing a database index value.

The rest of the paper is organized as follows, Section 2 presents a short review of lightweight mutual authentication protocols . After that, a full review of Yeh et al.'s protocol and its functionality is described in Section 3. Section 4 elaborates on the vulnerability analysis of the protocol and its assumptions. A de-synchronization attack is proposed in Sub-Section 4.1 and an impersonation attack is presented in Sub-Section4.2. Finally, conclusions are given in Section 5.

2. REVIEW OF LIGHTWEIGHT MUTUAL AUTHENTICATION PROTOCOLS

RFID tags compliant with EPC Class-1 Generation-2 (Gen-2 in shorts) are based on transponders with limited functulities; e.g. 16-bit pseudo-random number generator (PRNG), 16-bit cyclic redundancy check code (CRC), and bitwise operations such as XOR, AND, and OR [11].

Several protocols were proposed with the aim of securing Gen-2 tags. Unfortunately the majority of these protocols failed either to fulfill Gen-2 requirements or to satisfy the claimed security properties. For instance, [12] presented a protocol using a PIN password to secure the communication, however, it is shown to be volnurable to several attacks [13] [14]; First, it is vulnerable to a de-synchronization attack as a consequence of the weak updating mechanism of the secret keys and the shared values. Secondly, it does not offer protection against replay attacks and a passive attacker can reuse tokens from

previous sessions. Thirdly, it is susceptible to a traceability attack since tags respond with the same value every time – in this last one, the attacker has to intercept the updating message and the tag would respond with a constant value.

2.1. Yeh et al.'s Protocol

This protocol was initially designed to overcome security weaknesses of a previous protocol proposed by Chien and Chen's protocol [15]. In particular, the authors addressed DoS attacks, privacy concerns, and database computation overload. As its predecessor, the protocol avoids the usage of CRC functions due to its linearity. The tag stores K_i, P_i, C_i, EPC_s and the database keeps copy of the K_{old}, P_{old}, C_{old}, K_{new}, P_{new}, C_{new}, RID, EPCs and DATA values, and finally the reader stores into its memory the reader identification, named as RID (see Fig.1 for details).

The protocol is divided into two phases: initialization phase and authentication phase.

Initialization phase

For each tag, the server randomly generates an initial authentication key (K_0), initial access key (P_0), and database index (C_0). It sets these values for the records stored in the tag (i.e., $K_i=K_0$, $P_i=P_0$, $C_i=C_0$) and the corresponding records stored on the database (i.e., $K_{old}= K_0$, $K_{new}=K_0$, $P_{old}= P_0$, $P_{new}= P_0$, $C_{old}=C_0$ and $C_{new}=C_0$).

(i + 1)th authentication phase

1. R →Tag_x: N_R

The reader generates a nonce N_R and sends it to the tag as a challenge. Upon receiving N_R, the tag generates another random number N_T and uses both random values along with the values stored into its memory (K_i, P_i, C_i, EPC_s) to compute the values (M1, D, C_i, E) using the following formulas:

$M1= PRNG(EPC_S \oplus N_R) \oplus K_i$

$D = N_T \oplus K_i$

$E = N_T \oplus PRNG(C_i \oplus K_i)$

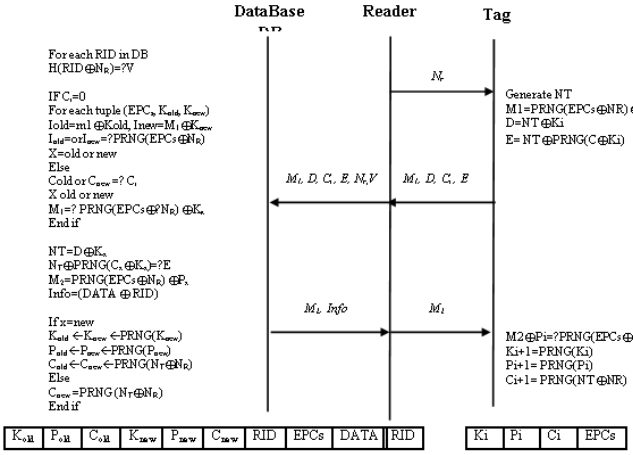

Fig.1. Yeh et al's Proposal

2. Tag →R: M_1, D, Ci, E

The tag forwards the values (M1, D, C_i, E) in one message to the reader.

3. R → DB: V, M1, D, C_i, E, N_R

The reader computes value V using the formula $V=H(RID \oplus N_R)$, then forwards it together with the contents of Message 1 (N_R) and Message 2 (M1, D, C_i, E) to the database. Once received, the database performs the following operations:

The database iteratively picks up each stored RID and computes $H(RID \oplus N_R)$ for that record to authenticate the reader based on received value V. If a match is found, the reader will be authenticated and the procedure will continue. Otherwise, the session will be terminated.

After the reader is authenticated, the database examines the value of C_i. If $C_i=0$, then this is a first time access. The database picks up every record sequentially, and computes three values (I_{old}, I_{new}, and $PRNG(EPC_S) \oplus N_R$) based on the received values M_1, N_R, and the stored ones (K_{old}, K_{new}, EPC_S), where $I_{old}= M_1 \oplus K_{old}$ and $I_{new}=M_1 \oplus K_{new}$. When a match is found, where $PRNG(EPCS) \oplus N_R=I_{new}$ or $PRNG(EPCS) \oplus N_R=I_{old}$, the database sets X as "new" for the first case, and "old" for the second case. Otherwise, if C_i< >0, then C_i is used as an index to look up the corresponding record in the database. The database checks the received value of C_i against the stored values for the corresponding record, where, if $C_i=C_{old}$ then the database marks X in K_x as "old". Contrarily, if $C_i=C_{new}$, then the database marks X in K_x as "new". The database subsequently recalculates M_1 based on K_x value.

The database uses the formula $N_T=D \oplus K_X$ to retrieve the value of N_T from the sub-message D. It then computes output value from the formula $N_T \oplus PRNG(C_X \oplus K_X)$ and compares this value with the sub-message E, which was received from the tag. If they do not match, the reader aborts the protocol and ends the session. Otherwise, the following applies:

4. DB → R: M_2, Info

The database computes the value of sub-message M_2 using the formula M_2=PRNG($EPC_S \oplus N_T$)$\oplus P_X$ and the sub-message *Info* using the formula Info=(DATA⊕RID). Subsequently, it forwards these two values (M_2, Info) to the reader. The database then updates the record's values (C_{old}, C_{new}, K_{old}, K_{new}, P_{old}, P_{new}) based on the identified X's value, where if X=old then C_{new}=PRNG($N_T \oplus N_R$) and the rest of the values remain unchanged. Otherwise, if X=new, then the record's values labeled as *new* in the current session *(C_{new}, K_{new}, P_{new})* becomes the *old* values for the next session (C_{old}, K_{old}, P_{old}) where K_{old}=K_{new}, P_{old}=P_{new}, and C_{old}=C_{new}. Values labeled as *new* are computed for the next session using the following formulas: K_{new}=PRNG(K_{new}), P_{new}=PRNG(P_{new}), and C_{new}=PRNG($N_T \oplus N_R$)

When the reader receives the message, it obtains DATA from the info field by inversing the formula DATA=info⊕RID using the RID stored in it. After that, it forwards M2 to the tag.

5. R →Tag: M_2

When M_2 is delivered, the tag computes PRNG($EPC_S \oplus N_T$) from its data and computes $M_2 \oplus P_i$ from the received M_2 and stored P_i. If they are equal, its values must be updated by K_{i+1}=PRNG(K_i), P_{i+1}=PRNG(P_i) from its stored values of K_i and P_i, respectively, and C_{i+1}=PRNG($N_T \oplus N_R$) by the N_R it received from the reader and N_T it generated at the beginning of the session. It then concludes the session. Otherwise, the protocol is aborted and old values are preserved.

3. VULNERABILITY OF YEH ET AL.'S PROTOCOL

In this paper we present a de-synchronization and impersonation attacks against Yeh et al.'s protocol. These attacks complement the ones (integrity and forward security problems) presented in [16].

3.1. De-synchronization attack

Yeh et al.'s protocol was designed using two sets of authentication and access keys to combat DoS attack, which causes a de-synchronization state between the tag and the server. The authors criticized the fact that its predecessor scheme (i.e., Chien and Chen's protocol [15]) updated the key values (K_{old} and P_{old}) on every successful mutual authentication session at the database side. Motivated by this, Yeh et al. proposed to add a validation criterion for this updating mechanism to solve the de-synchronization attack, which Chien and Chen's protocol suffers from, and is based on the usage of the new values D, E, and C_i. Nevertheless, despite these validation tokens, we show how replay attacks can de-synchronize the protocol. The used adversary (malicious reader) has to be able to interrupt and forward messages only, and it does not need to have the capability to communicate with the database. This adversary will execute two session procedures in one session. That is, both communication sessions are executed almost in parallel but with only a slight difference in time:

In the $(i+1)^{th}$ authentication session, the malicious reader will intercept the last message from the database and throw away M_2 message to keep the tag using the same index value C_{i+1}. At the same time, the database will update its local parameters, specifically C_{old} would be C_{new}, and C_{i+1}, and its C_{new} would be C_{i+2}.

In a slightly posterior session (almost a parallel session), the malicious reader will resend a new Message 3. However, instead of containing (V, M1, D, C_i, E, N_R), it will send (V, M1, D⊕RND, C_i, E⊕RND, N_R), which will allow the database to understand that it is a new session. These values (i.e., V, M1, D⊕RND, C_i, E⊕RND, N_R) will facilitate the tag to be authenticated by the database because N_R continues to represent the same values from the eavesdropped session. N_T will become NT⊕RND, which is used correctly in D and E messages. Due to modified Message 3 sent by the reader, the database will update its C_{new} value based on the C_x (in this case, X=old) from C_{i+2} to C_{i+3}. At the same time, the malicious reader will forward the stored M2 message to the tag, causing the tag to update its values from (K_{i+1}, P_{i+1} and C_{i+1}) to (K_{i+2}, P_{i+2} and C_{i+2}).

At this step the tag will store C_{i+2} as index value, and the database will keep the values C_{i+1} and C_{i+3}. Therefore, the tag and the database lost its synchronization and this is permanent. In fact, the tag can never be identified because the search index stored into its memory is different from the two index (old and new) values stored in the database.

3.2. Impersonation attack

In this section, we introduce tag impersonation attack conducted by a dishonest reader. The key points of this attack are based on the use of N_T nonce in both D and E tokens and the abusive use of the bitwise XOR operations. Bitwise operations like XOR are linear functions, which are vulnerable to active and passive attacks. The proposed attack is sketched below:

$(i+1)^{th}$ authentication phase

(1) R → Tag_x: N_R
(2) Tag → R: M1, D, C_i, E
- M1= PRNG ($EPC_S \oplus N_R$)$\oplus K_i$
- D = $N_T \oplus K_i$
- E = $N_T \oplus$PRNG($C_i \oplus K_i$)

(3) R → DB: V, M1, D, C_i, E, N_R
(4) DB → R: M2, Info
(5) R → Tag_x: Attack

The attack can be performed using two methods. First by preventing the reader from forwarding any messages to the tag. Alternatively, the adversary can interrupt the last message and send a fraudulent message containing an incorrect value of M2.

At this point, the targeted tag is isolated and the malicious reader can replace and impersonate the original tag by computing simple bitwise XOR operations as described in the following.

- $(i + n)^{th}$ authentication phase (n>2)

Basically the fraudulent reader simulates that the tag always incorrectly receives the message M2. Therefore, the updating phase is not run in the tag and previous M1 message is valid.

In detail, M1, D, E, N_R, and V are the picked values of a previous legitimate session. After the reception of M2, the reader block this message and simulates the tag incorrectly received M2. After that, the fraudulent reader sends `M1, D⊕RND, E⊕RND, N_R, V,` where `RND` represents an arbitrary random value. The tag is authenticated since `M1` is legitimate. The random number N_T' associated to this session is the bitwise XOR between N_T and RND. We sketch the process below:

DB→R: M2, Info `Fake R →DB: M1, D⊕RND, E⊕RND, N_R, V`

The proposed attack can be executed indefinitely as the original scheme does not assume any threshold for the number of times the M2 message can be interrupted, altered, or incorrectly received.

4. CONCLUSION

We scrutinize the security of Yeh et al.'s protocol for RFID Gen-2 tags. The security properties claimed by the authors in the original protocol have been refuted by proposing attacks against the authentication scheme. We show how the protocol is vulnerable against de-synchronization and impersonation attacks. The attacks can be conducted by a malicious reader, which mainly forwards message and does simple modifications exploiting the weaknesses of the bitwise XOR operations. Complementary to the attacks presented in this paper, in [16] Yoon showed how the scheme suffers from integrity and forward secrecy problems.

The design of secure RFID authentication protocols compliant with Gen2 standard is still an open-challenge. New proposals have to be presented together with a rigorous security analysis to avoid such trivial security pitfalls.

REFERENCES

[1] Kim, M., et al., Forward-backward analysis of RFID-enabled supply chain using fuzzy cognitive map and genetic algorithm. Expert Systems with Applications, 2008. **35**(3): p. 1166-1176.

[2] Sun, Q., H. Zhang, and L. Mo, Dual reader wireless protocols for dense active RFID identification. International Journal of Communication Systems, 2011. **24**(11): p. 1431-1444.

[3] Cho, K., et al., An extensible and ubiquitous RFID management framework over next generation network. International Journal of Communication Systems, 2009. **23**(910): p. 1093-1110.

[4] Kim, S.C., J.S. Cho, and S.K. Kim, Performance improvement of hybrid tag anti collision protocol for radio frequency identification systems. International Journal of Communication Systems, 2012.

[5] Weis, S., et al., Security and privacy aspects of low-cost radio frequency identification systems. Security in Pervasive Computing, 2004: p. 50-59.

[6] Juels, A., RFID security and privacy: A research survey. IEEE Journal on Selected Areas in Communications, 2006. **24**(2): p. 381-394.

[7] Lim, C. and T. Korkishko, mCryptontion A lightweight block cipher for security of low-cost RFID tags and Sensors. Information Security Applications, 2006: p. 243-258.

[8] Li, J.S. and K.H. Liu, A hidden mutual authentication protocol for low cost RFID tags. International Journal of Communication Systems, 2011. **24**(9): p. 1196-1211.

[9] Yeh, T., et al., Securing RFID systems conforming to EPC Class 1 Generation 2 standard Expert System With Application 2010. **37**(4): p. 7678-7683.

[10] Chien, H., SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing, 2007. **4**(4): p. 337-340.

[11] EPCglobal. Class 1 Generation 2 UHF Air Interface Protocol Standard "Gen 2" Version 1.2.0. 2008 [cited 2010 15th October]; Available from: http://www.epcglobalinc.org/standards/.

[12] Chien, H., SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transactions on Dependable and Secure Computing, 2007. **4**(4): p. 337-340.

[13] Duc, D., et al. Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. in The Symposium on Cryptography and Information Security, 2006. 2006: Citeseer.

[14] Sun, H.-M., W.-C. Ting, and K.-H. Wang, On the Security of Chien's Ultra-Lightweight RFID Authentication Protocol. IEEE Transactions on Dependable and Secure Computing, 2009. **99**.

[15] Chien, H.Y. and C.W. Huang, Security of ultra-lightweight RFID authentication protocols and its improvements. ACM SIGOPS Operating Systems Review, 2007. **41**(4): p. 86.

[16] Chien, H. Y. and Chen, C. H. Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards. Computer Standards and Interfaces, 2207. 29: p. 254-259.

[17] Yoon, E.J. Improvement of the Security systems conforming to EPC Class 1 Generation 2 standard. Expert Systems and Applications, 2012. 39: p. 1589-1595.