Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard

Mohammad Hassan Habibi, Mahdi R. Alagheband, Mohammad Reza Aref

Attacks on a Lightweight Mutual Authentication Protocol Under EPC C-1 G-2 Standard

Mohammad Hassan Habibi, Mahdi R. Alagheband, Mohammad Reza Aref

EE Department, ISSL Laboratory, Sharif University of Technology, Tehran, Iran
{mohamad.h.habibi@gmail.com, m.alaghband@srbiau.ac.ir, aref@sharif.edu}

Abstract. Yeh et al. have recently proposed a mutual authentication protocol based on EPC Class-1 Gen.-2 standard. They claim their protocol is secure against adversarial attacks and also provides forward secrecy. In this paper we show that the proposed protocol does not have cited security features properly. A powerful and practical attack is presented on this protocol whereby the whole security of the protocol is broken. Furthermore, Yeh et al.'s protocol does not assure the *untraceabilitiy* and *backward untraceabilitiy* attributes. We also will propose our revision to safeguard the Yeh et al.'s protocol against cited attacks.

Keywords: RFID, authentication, EPC C-1 G-2 standard, Security analysis, Traceability attack.

1 Introduction

Nowadays Radio Frequency Identification (RFID) technology has been incorporated in our daily life and employed in many applications e.g. public transportation passes [1], supply chain management [2], e-passport [3] etc. RFID systems include tags, readers and back-end server. The tag is a low cost device with a constraint microchip, small memory and antenna to communicate with the reader. The readers are placed between tags and back-end server as an intermediary for message transmission. Not surprisingly, the back-end server has the whole information and secret values of all tags.

EPC Class-1 Gen.-2 standard is a framework for RFID communications, defined by EPC global (Electronic Product Code) organization [4, 5] but RFID authentication protocols based on it have undergone noticeable difficulties to satisfy the perfect security characteristics.

In order to have secure authentication protocols, an adversary should not be able to obtain any information about the target tag. Privacy and untraceability are two important issues relevant to RFID systems. Thus, an authentication protocol should assure the privacy characteristics including *untraceability* and *backward untraceability* for tags and their holders [6]. On the other side, RFID authentication protocols are under different threats, defined as follows.

Information leakage: the tag and reader perform an authentication protocol and exchange some messages with each other. Since the wireless communication channel

is insecure, it can be eavesdropped by an adversary. Hence, each authentication protocol should be designed in a way that the adversary, with reasonable computational capabilities, does not be able to exploit the exchanged messages [7].

Tag Tracing and tracking: Tag tracing and tracking are damaging problems in RFID systems. Even when the leakage of information is impossible, the untraceability of tag and its holder is not guaranteed in RFID systems. Untraceability means that if an adversary eavesdrops message transmission between a target tag and a reader at time t, he does not be able to distinguish an interaction of that tag at time $t' > t$ [8].

DoS attack: denial-of-Service (*DoS*) is another attack on RFID systems. An adversary tries to find ways to fail target tag from receiving services, e.g. in the desynchronization attack, as one kind of *DoS* attacks, the shared secret value between the tag and the back-end server is made inconsistent by an adversary. Then, the tag and back-end server cannot recognize each other in future and tag becomes disabled [9].

Many RFID authentication protocols have been proposed [10, 11, 12, 13, 14, 15]. Although these protocols tried to provide secure and untraceable communication for RFID systems, however many weaknesses have been found in them [16, 17, 18, 19, 20, 21]. In this context, Yeh et al. have recently proposed a RFID mutual authentication protocol compatible with EPC C-1 G-2 standard [22] that we name SRP (Securing RFID Protocol) in this paper. The authors have claimed that not only SRP does not reveal any information but also it has forward secrecy and robustness against *DoS* attack. In this paper, we prove that SRP is vulnerable to a powerful and fatal attack that needs only 2^{16} off-line PRNG (pseudo random number generator) computations. Furthermore, the whole security of this protocol will be destroyed inasmuch as the RFID system is most vulnerable to tag and reader impersonation, *DoS* attack, *untraceability* and *backward untraceability*. Finally we propose our revision to prevent the mentioned attacks.

2 Review SRP

2.1 Initialization phase

The nine secret values K_{old}, P_{old}, C_{old}, K_{new}, P_{new}, C_{new}, EPC_s, RID and DATA corresponding to each tag is loaded in database. Besides, random values K_0, P_0 and C_0 are generated by manufacturer and the recorded values are set in a way that K_{old}=K_{new}=K_0, P_{old}=P_{new}=P_0 and C_{old}=C_{new}=C_0. Each tag records four values K_i=K_0, P_i=P_0, C_i=C_0 and EPC_s.

2.2 The (i+1)th Authentication Round

In this part, the SRP protocol is briefly described. The following steps explain the protocol in the round ($i+1$).

1. The reader generates number N_R randomly and sends it to the tag.
2. Receiving N_R, the tag generates random number N_T and computes:

M1=PRNG($EPC_s \oplus N_R$)$\oplus K_i$, $D=N_T \oplus K_i$ and $E=N_T \oplus$PRNG($C_i \oplus K_i$). Then the tag forwards (C_i, *M1, D, E*) to the reader.

3. The reader computes *V*=H($RID \oplus N_R$) and sends (C_i,*M1,D,E,* N_R,*V*) to the database.

Receives (C_i, *M1, D, E,* N_R, *V*), the database performs the following procedure:

a) For each stored *RID*, computes H($RID \oplus N_R$) and compares it with *V* to find whether the computed value is equal to *V*. If it is true, the database will authenticate the reader.

b) Based on value C_i, one of the two following procedures is occurred:

i. The database computes PRNG ($EPC_s \oplus N_R$), I_{old}=*M1*$\oplus K_{old}$ and I_{new}=*M1*$\oplus K_{new}$ provided that C_i = 0, because it means the first access. Then it checks whether I_{old} or I_{new} correspond to PRNG($EPC_s \oplus N_R$). This process is regularly repeated until a match equality is founded. X is set to either *old* or *new* provided that either I_{old} or I_{new} is the match, respectively.

ii. If $C_i \neq 0$, the database uses C_i as an index to find the corresponding recorded entry. When the database finds an entry correspondent to C_i, then the value of *X* is determined either *old* or new provided that $C_i = C_{old}$ or C_{new} respectively. Corresponding K_X and EPC_s are extracted to check whether PRNG($EPC_s \oplus N_R$)$\oplus K_X$ is equal to *M1* or not.

The database obtains N_T with the aid of K_X and D, and ensures whether $N_T \oplus$PRNG($C_X \oplus K_X$) is equal to the received *E*.

c) Computes *M2*=PRNG($EPC_s \oplus N_T$)$\oplus P_X$ and *Info*=(*DATA*$\oplus$*RID*), and sends them to the reader.

d) If $X = new$, it updates the stored values as follows: $K_{old}=K_{new}$, K_{new}=PRNG(K_{new}), $P_{old}=P_{new}$, P_{new}=PRNG(P_{new}), $C_{old} = C_{new}$, C_{new}=PRNG($N_T \oplus N_R$). But if *X*=*old*, it just updates C_{new} as C_{new}=PRNG($N_T \oplus N_R$).

4. The reader does XOR operation with *RID* and the received *Info* and extracts *DATA*, and sends *M2* to the tag. The tag picks up the stored P_i and computes $P_i \oplus$*M2* to find whether it is equal to PRNG($EPC_s \oplus N_T$). If the matching is found, the database is authenticated and the tag updates as follows: K_{i+1}=PRNG (K_i), P_{i+1}=PRNG (P_i), C_{i+1}=PRNG ($N_T \oplus N_R$).

3 Vulnerabilities of SRP

In this section we show the vulnerabilities of SRP. First a practical and powerful attack on SRP is presented. Then, we show that an adversary obtains the most important secret value of a tag which called EPC_s, and show that SRP is vulnerable to tracing attacks. Hence, we show that the SRP does not provide *backward untraceability* and *untraceability*.

3.1 Reveal EPC_s

Since N_R and N_T are XORed with EPC_s, we can conclude the N_R and N_T bit lengths are the same as EPC_s bit length. Furthermore, K_i, P_i and C_i bit length must be

equal to the PRNG bit length inasmuch as they are updated by PRNG. Due to the fact that the EPC_s bit length is very short and fix in all rounds of the SRP, an adversary can exploit this subject to get EPC_s. He just needs to perform two consecutive sessions with the target tag and calculate 2^{16} off-line PRNG computations. The procedure of our attack is explained as follows.

1. The adversary starts a session with the target tag $\mathcal{T}_i$ in the round ($i+1$) by sending random number N_{R1} and $\mathcal{T}_i$ replies with (C_i, $M1_1$, D_1, E_1). The adversary reserves $M1_1$ and terminates the session. He performs the second session with $\mathcal{T}_i$ by transmission of N_{R2} and gets tag's response as (C_i, $M1_2$, D_2, E_2).
2. Since the first session is not completed, $\mathcal{T}_i$ does not update its secret key K_i for the second session. Hence $M1_1$ and $M1_2$ are constructed as follows:
   $M1_1$ =PRNG(EP$C_s \oplus N_{R1}$) $\oplus K_i$, $M1_2$ =PRNG(EP$C_s \oplus N_{R2}$)$\oplus K_i$.
3. $\mathcal{A}$ omits K_i by XORing $M1_1$ and $M1_2$: $M1_1 \oplus M1_2$=PRNG(EP$C_s \oplus N_{R1}$)$\oplus K_i \oplus$ PRNG(EP$C_s \oplus N_{R2}$)$\oplus K_i$=PRNG(EP$C_s \oplus N_{R1}$)$\oplus$PRNG(EP$C_s \oplus N_{R2}$)=β, Where β is a 16-bit string as a result of $M1_1 \oplus M1_2$.
4. Let L=$\{l_1, l_2, \ldots, l_{2^{16}}\}$ be the set of all bit strings with length 16. Since EPC_s is a bit string with length 16, $EPC_s \in$L. Therefore, the adversary with the aid of β, N_{R1} and N_{R2}, executes below algorithm to reach correct EPC_s the adversary proceeds according to the below algorithm:

   Algorithm 1

   For $1 \leq i \leq 2^{16}$
   Choose $l_i \in$ L
   α =PRNG($l_i \oplus N_{R1}$)$\oplus$PRNG($l_i \oplus N_{R2}$), If $\alpha = \beta$ then return l_i as EPC_s
   End for

After at most 2^{16} execution of the algorithm, the adversary finds the correct EPC_s. As a result of the above attack, we present three noticeable attacks on SRP including tag impersonation, reader impersonation and *DoS* attack.

3.1.1 Tag Impersonation

An adversary simply gets the secret key K_i by a passive attack. Indeed, he listens to the communication channel between the legitimate reader $\mathcal{R}$ and the target tag $\mathcal{T}_i$ in the round ($i+1$) to obtain N_{R3} and (C_i, $M1_3$, D_3, E_3). Since the adversary has EPC_s, he computes PRNG ($EPC_s \oplus N_{R3}$). Thus the secret key K_i is computed as: K_i=$M1_3 \oplus (EPC_s \oplus N_{R3})$ and K_{i+1}=PRNG(K_i). The random number N_{T3} is computed as: N_{T3}=$D \oplus K_i$ and finally the index for the next session is computed as C_{i+1}=PRNG($N_{T3} \oplus N_{R3}$).

Now, the adversary starts a new session with the reader. $\mathcal{R}$ sends N_{R4} to him and he replies (C_i, $M1_4$, D_4, E_4) where $M1_4$=PRNG($EPC_s \oplus N_{R4}$)$\oplus K_i$, D_4=$N'_{T4} \oplus K_i$ and E_4=$N'_{T4}$$\oplus$PRN($C_i \oplus K_i$). Since these values are correctly computed, the database accepts the adversary and authenticates him.

3.1.2 Reader Impersonation and DoS Attack

SRP is also vulnerable by two other attacks. By revealing EPC_s, the adversary can forge a legitimate reader and then desynchronize the target tag. The procedure of these attacks is explained as follows.

1. The adversary listens to the communication between $\mathcal{R}$ and $\mathcal{T}_i$ in the round ($i+1$) to obtain N_{R5}, (C_i, $M1_5$, D_5, E_5) and $M2_5$. As the adversary has EPC_s, he computes PRNG($EPC_s \oplus N_{R5}$) and gets the secret key K_i as: K_i=$M1_5 \oplus$PRNG($EPC_s \oplus N_{R5}$) and K_{i+1}=PRNG(K_i). The secret key P_i is gotten as: P_i=$M2_5 \oplus$PRNG($EPC_s \oplus N_{T5}$) and P_{i+1}=PRNG(P_i) where N_{T5}=$D_5 \oplus K_i$.
2. He begins a new session with $\mathcal{T}_i$ and sends N_{R6} to it. $\mathcal{T}_i$ replies with (C_{i+1}, $M1_6$, D_6, E_6), created by EPC_s, N_{R6}, K_i, N_{T6} and C_{i+1}.
3. After receiving the tag's response, the adversary extracts N_{T6} (N_{T6}=$D_6 \oplus K_i$), computes $M2_5$=PRNG($EPC_s \oplus N_{T6}$)$\oplus P_{i+1}$ and sends it to the tag.
4. $\mathcal{T}_i$ checks whether $M2_5 \oplus P_{i+1}$ is equal to PRNG($EPC_s \oplus N_{T6}$) or not. $\mathcal{T}_i$ authenticates the adversary and updates its secret values provided that the equation will be true: K_{i+2}=PRNG(K_{i+1}), P_{i+2}=PRNG(P_{i+1}), C_{i+2}=PRNG($N_{R6} \oplus N_{T6}$). Eventually, the stored secret values on $\mathcal{T}_i$ are (K_{i+2}, P_{i+2}, C_{i+2}, EPC_s) whereas the database has stored (K_i, P_i, C_i, K_{i+1}, P_{i+1}, C_{i+1}, *RID*, EPC_s, *DATA*). Therefore, the tag and reader have been desynchronized because the secret stored values in database are completely different from the values stored in the tag.

3.2 Privacy Analysis

The authors of SRP have specified that not only their protocol have forward secrecy, but also SRP is resistant to the tracing attacks. We show that SRP does not have forward secrecy and we also present a *traceability* attack on SRP.

3.2.1 Privacy Model

There are privacy models for the evaluation of RFID protocols [6, 23, 24, 25, 26]. We analyze SRP protocol based on Ouafi and Phan model [26] which is based on [24] and [6]. The model is summarized as follows.

The protocol parties are tags ($\mathcal{T}$) and readers ($\mathcal{R}$) which interact in protocol sessions. In this model an adversary $\mathcal{A}$ controls the communication channel between all parties by interacting either passively or actively with them. The adversary $\mathcal{A}$ is allowed to run the following queries:

- **Execute** ($\mathcal{R}$, $\mathcal{T}$, i) query. This query models the passive attacks. The adversary $\mathcal{A}$ eavesdrops on the communication channel between $\mathcal{T}$ and $\mathcal{R}$ and gets read access to the exchanged messages between the parties in session i of a truthful protocol execution.
- **Send** ($\mathcal{U}$, $\mathcal{V}$, m, i) query. This query models active attacks by allowing the adversary $\mathcal{A}$ to impersonate some reader $\mathcal{U} \in \mathcal{R}$ (respectively tag $\mathcal{V} \in \mathcal{T}$) in some protocol session i and send a message m of its choice to an instance of some tag $\mathcal{V} \in \mathcal{T}$ (respectively reader $\mathcal{U} \in \mathcal{R}$). Furthermore the adversary $\mathcal{A}$ is allowed

to block or alert the message m that is sent from $\mathcal{U}$ to $\mathcal{V}$ (respectively $\mathcal{V}$ to $\mathcal{U}$) in session i of a truthful protocol execution.

- **Corrupt** ($\mathcal{T}$, K') query. This query allows the adversary $\mathcal{A}$ to learn the stored secret K of the tag $\mathcal{T} \in \mathcal{T}$, and which further sets the stored secret to K'. **Corrupt** query means that the adversary has physical access to the tag, i.e. the adversary can read and tamper with the tag's permanent memory.
- **Test** (i, $\mathcal{T}_o$, $\mathcal{T}_1$) query. This query does not correspond to any of $\mathcal{A}$'s abilities, but it is necessary to define the untraceability test. When this query is invoked for session i, a random bit $b \in \{0, 1\}$ is generated and then, $\mathcal{A}$ is given $\mathcal{T}_b \in \{\mathcal{T}_o, \mathcal{T}_1\}$. Informally, $\mathcal{A}$ wins if he can guess the bit b.

Untraceable privacy (*UPriv*) is defined using the game $\mathcal{G}$ played between an adversary $\mathcal{A}$ and a collection of the reader and the tag instances. The game $\mathcal{G}$ is divided into three following phases:

- **Learning phase:** $\mathcal{A}$ is given tags $\mathcal{T}_o$ and $\mathcal{T}_1$ randomly and he is able to send any **Execute**, **Send** and **Corrupt** queries of its choice to $\mathcal{T}_0$, $\mathcal{T}_1$ and reader.
- **Challenge phase:** $\mathcal{A}$ chooses two fresh tags $\mathcal{T}_0$, $\mathcal{T}_1$ to be tested and sends a **Test** (i, $\mathcal{T}_o$, $\mathcal{T}_1$) query. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$.$\mathcal{A}$ continues making any **Execute**, and **Send** queries at will.
- **Guess phase:** finally, $\mathcal{A}$ terminates the game $\mathcal{G}$ and outputs a bit $b' \in \{0, 1\}$, which is its guess of the value of b.

The success of $\mathcal{A}$ in winning game $\mathcal{G}$ and thus breaking the notion of *UPriv* is quantified in terms $\mathcal{A}$ advantage in distinguishing whether $\mathcal{A}$ received $\mathcal{T}_0$ or $\mathcal{T}_1$ and denoted by $\mathbf{Adv}_{\mathbf{A}}^{\mathbf{UPriv}}(k)$ where k is the security parameter.

$$\mathbf{Adv}_{\mathbf{A}}^{\mathbf{UPriv}}(k) = |\Pr(b = b') - \Pr(\text{random flip coin})| = |\Pr(b' = b) - \tfrac{1}{2}| \quad \text{where}$$
$$0 \leq \mathbf{Adv}_{\mathbf{A}}^{\mathbf{UPriv}}(k) \leq \tfrac{1}{2}.$$

Besides, the notion *backward untraceability* is defined as: "*backward untraceability* states that even if given all the internal states of a target tag at time t, the adversary shouldn't be able to identify the target tag's interactions that occur at time t' < t" [6].

3.2.2 Backward traceability

In this section we show how to break the notion *backward untraceability* in the SRP protocol. Because EPC_s is constant in the all rounds of SRP, an adversary $\mathcal{A}$ can track the target tag with doing the following steps:

- **Learning phase:** $\mathcal{A}$ sends a **Corrupt** ($\mathcal{T}_0$, K') query in the round (i+1) and obtains $(K_i^{T_0}, P_i^{T_0}, C_i^{T_0}, EPC_{s,i}^{T_0})$.
- **Challenge phase:** $\mathcal{A}$ chooses two fresh tags ($\mathcal{T}_0$, $\mathcal{T}_1$) to be tested and sends a **Test** (i, $\mathcal{T}_o$, $\mathcal{T}_1$) query. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$. $\mathcal{A}$ makes an **Execute** ($\mathcal{R}$, $\mathcal{T}_b$, i) query in the round (i) and as a result, $\mathcal{A}$ is given messages $\{N_{R,i-1}^{T_b}, (M1_{i-1}^{T_b}, D_{i-1}^{T_b}, C_{i-1}^{T_b}, E_{i-1}^{T_b})\}$.

- **Guess phase:** finally, $\mathcal{A}$ terminates the game $\mathcal{G}$ and outputs a bit $b' \in \{0, 1\}$ as its guess of the value of b. In particular, $\mathcal{A}$ performs the following procedure to obtain the value b':

1. He computes PRNG($EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_b}$)$\oplus M1_{i-1}^{T_b} = \theta$ where θ is a 16-bit string.
2. $\mathcal{A}$ utilizes the following simple decision rule:

$$b' = \begin{cases} if\ D_{i-1}^{T_b} \oplus E_{i-1}^{T_b} = \theta \oplus \mathrm{PRNG}\left(C_{i-1}^{T_b} \oplus \theta\right) & b' = 0 \\ otherwise & b' = 1 \end{cases}$$

Hence we have:

$$\mathbf{Adv}_{\mathbf{A}}^{\mathbf{UPriv}}(k) = |\Pr(b' = b) - \Pr(\text{random flip coin})| = |\Pr(b' = b) - \tfrac{1}{2}| = |1 - \tfrac{1}{2}| = \tfrac{1}{2}$$

Proof: By the fact that EPC_s is a permanent value in the all rounds of the protocol, we have $\mathrm{EPC}_{s,i}^{T_0} = \mathrm{EPC}_{s,i-1}^{T_0}$. Thus we have the following procedure:

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow \mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_b}) = \mathrm{PRNG}(EPC_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_0}) \quad (1)$$

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow \mathrm{M1}_{i-1}^{T_b} = \mathrm{M1}_{i-1}^{T_0} = \mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_0}) \oplus \mathrm{K}_{i-1}^{T_0} \quad (2)$$

$$(1), (2) \Rightarrow \mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_b}) \oplus \mathrm{M1}_{i-1}^{T_b} = \mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_0}) \oplus \mathrm{M1}_{i-1}^{T_0} =$$

$$\mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_0}) \oplus \mathrm{PRNG}(\mathrm{EPC}_{s,i}^{T_0} \oplus \mathrm{N}_{R,i-1}^{T_0}) \oplus \mathrm{K}_{i-1}^{T_0} = \mathrm{K}_{i-1}^{T_0} = \theta \quad (3)$$

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow D_{i-1}^{T_b} \oplus E_{i-1}^{T_b} = D_{i-1}^{T_0} \oplus E_{i-1}^{T_0} = N_{\mathrm{T},i-1}^{T_0} \oplus \mathrm{K}_{i-1}^{T_0} \oplus N_{\mathrm{T},i-1}^{T_0} \oplus \mathrm{PRNG}(\mathrm{C}_{i-1}^{T_0} \oplus \mathrm{K}_{i-1}^{T_0}) =$$

$$\mathrm{K}_{i-1}^{T_0} \oplus \mathrm{PRNG}(\mathrm{C}_{i-1}^{T_0} \oplus \mathrm{K}_{i-1}^{T_0}) = \theta \oplus \mathrm{PRNG}(\mathrm{C}_{i-1}^{T_0} \oplus \theta) = \theta \oplus \mathrm{PRNG}(C_{i-1}^{T_b} \oplus \theta) \quad (4)$$

▪

3.2.3 Traceability attack

An authentication protocol for RFID systems should assure the privacy of a tag and its holder. However, many RFID protocols put it at risk by designing protocols where tags answer reader's queries with permanent values. Thus performing traceability attacks not only possible but trivial.

Now, we prove the SRP does not guarantee privacy location and allows tags tracking.

- **Learning phase:** $\mathcal{A}$ sends an **Execute** ($\mathcal{R}$, $\mathcal{T}_0$, i+1) query in the (i+1)*th* round by sending N_{R1} and obtains ($M1_i^{T_0}, D_i^{T_0}, C_i^{T_0}, E_i^{T_0}$).
- **Challenge phase:** $\mathcal{A}$ chooses two fresh tags ($\mathcal{T}_0$, $\mathcal{T}_1$) to be tested and sends a **Test** (i+1, $\mathcal{T}_0$, $\mathcal{T}_1$) query. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$. $\mathcal{A}$ makes an **Execute** ($\mathcal{R}$, $\mathcal{T}_b$, i+1) query by sending N_{R1} and as a result, $\mathcal{A}$ is given messages ($M1_i^{T_b}, D_i^{T_b}, C_i^{T_b}, E_i^{T_b}$).
- **Guess phase:** finally, $\mathcal{A}$ terminates the game $\mathcal{G}$ and outputs a bit b' $\in \{0, 1\}$ as its guess of the value of b. In particular, $\mathcal{A}$ utilizes the following simple decision rule:

$$b' = \begin{cases} \text{if } M1_i^{T_b} = M1_i^{T_0} & b' = 0 \\ \text{otherwise} & b' = 1 \end{cases}$$

Hence we have:

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

Proof: According to the protocol, we have the following equations:

$M1_i^{T_0}$=PRNG($EPC_{s,i}^{T_0} \oplus N_{R1}$)$\oplus$ $K_i^{T_0}$ (5)

$M1_i^{T_b}$= PRNG($EPC_{s,i}^{T_b} \oplus N_{R1}$)$\oplus$ $K_i^{T_b}$ (6)

Note that $\mathcal{T}_0$ does not update its secrets in the **Learning phase** and uses the same secret key K_i in both **Learning** and **Challenge phase**. Now we have the following result:

If $\mathcal{T}_b = \mathcal{T}_0$ $\Rightarrow M1_i^{T_b}$=PRNG($EPC_{s,i}^{T_b} \oplus$NR1)$\oplus K_i^{T_b}$=PRNG($EPC_{s,i}^{T_0} \oplus$NR1)$\oplus K_i^{T_0}$

$=M1_i^{T_0}$ (7)

▪

4 Revised Protocol

In order to eliminate the mentioned vulnerabilities in 3.1 and 3.2 subsections, we can modify the message *M1* as: *M1*=PRNG($EPC_S \oplus N_R \oplus$Pi)$\oplus K_i$. Although the cited vulnerabilities are fixed by the above modification, the traceability problem still will be unsolved. Hence, we need to construct the message *M1* as following: *M1*=PRNG($EPC_S \oplus N_R \oplus N_T$)$\oplus K_i$ to provide a secure protocol against all cited attacks.

4.1 Security analysis

Now, we analyze the security of the revised protocol as following.

Untraceability: Due to the fact that N_T is a random and fresh value, the tag′s responses are different whenever an adversary sends query and therefore, the adversary is unable to trace a tag.

Backward untraceability: If an adversary knows EPC_s and N_R in worth case, he cannot recognize any previous interactions by a tag inasmuch as he does not know N_T.

Reveal EPC_s: Since EPC_s is constant and its length is short, the mentioned attacks in 3.1 subsection happened successfully. We have added the random and fresh value N_T in construction of *M1* to remove these flaws. As a result, when an adversary wants to reveal EPC_s, he has to perform 2^{48} calculations rather than 2^{16}.It is a noticeable improvement in SRP security.

5 Conclusion

In this paper, the significant security flaws in the Yeh et al. mutual authentication protocol were showed. We presented a powerful and practical attack on SRP which reveals the permanent secret value of the target tag. This attack leads to tag and reader impersonation and desynchronization attack on the protocol. Moreover, we proved that this protocol did not provide *untraceability* and *backward untraceability*. Our

privacy analysis has been presented in a formal privacy model. Finally, to eliminate all cited vulnerabilities, we revised the SRP protocol and constructed the message M1 in a new way.

Acknowledgment

This work was partially supported by Iran National Science Fund (INSF)-cryptography chair and research institute for ITC, Tehran, Iran.

References

[1] Transport for London, Oyster card, http://www.oystercard.co.uk.
[2] "Michelin Embeds RFID Tags in Tires". RFID Journal. http://www.rfidjournal.com/article/articleview/269/1 /1/. Accessed 17 Jan 2003
[3] Hoepman, J.-H., Hubbers, E., Jacobs, B., Oostdijk, M., Scherer, R.W.: Crossing borders: Security and privacy issues of the European e-passport. NAME (IWSEC 2006). LNCS, Springer-Heidelberg, vol. 4266 (2006) 152–167.
[4] EPCglobal Inc., http://www.epcglobalinc.org/.
[5] EPCglobal Inc., EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocols for Communications at 860 MHz – 960 MHz version 1.1.0, Available at [4].
[6] Lim, C.H., & Kwon, T.: Strong and robust RFID authentication enabling perfect ownership transfer. P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 1–20, 2006.
[7] Van Deursen, T., Radomirovic, S.: Attacks on RFID protocols. Cryptology ePrint Archive, Report 2008/310,2008. <http://eprint.iacr.org/>.
[8] Ouafi, K., and Phan, R. C.-W.: Traceable privacy of recent provably-secure RFID protocols. Proc. Sixth Int'l Conf. Applied Cryptography and Network Security (ACNS '08), pp. 479-489, 2008.
[9] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., and Ribagorda, A.: Vulnerability analysis of RFID protocols for tag ownership transfer. Computer Networks 54 (2010) 1502–1508.
[10] Chien, H., Chen, C.: Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards. Computer Standards & Interfaces, 29 (2007) 254–259.
[11] Konidala, D.M., Kim, Z., Kim, K.: A simple and cost-effective RFID tag-reader mutual authentication scheme. In: Proceedings of Int'l Conference on RFID Security (RFIDSec)'07, (2007) 141–152.
[12] Kulseng, L., Yu, Z., Wei, Y., and Guan, Y.: Lightweight mutual authentication and ownership transfer for RFID Systems. In: Proceedings of IEEE INFOCOM 2010, 1-5, 2010.
[13] Chien, H. Y.: 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.
[14] Song, B., and Mitchell, C. J.: RFID authentication protocol for low-cost tags. In: Proc. of Wisec'08, pp.140-147, 2008.
[15] Duc, D.N., Park, J., Lee, H., Kim, K.: Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. The Symposium on Cryptography and Information Security (2006).
[16] Han, D., Kwon, D.: Vulnerability of an RFID authentication protocol conforming to EPC Class-1Generation-2 Standards. Computer Standards & Interfaces 31 (2009) 648–652.

[17] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., and Ribagorda, A.: Practical attacks on a mutual authentication scheme under the EPC Class-1 Generation-2 standard. Computer Communications 32 (2009) 1185–1193.

[18] Habibi, M. H., Gardeshi, M., Alagheband, M.R.: Attacks and improvements to a new RFID Authentication protocol. In proceedings of Third Workshop on RFID Security: RFIDsec Asia 2011, China.

[19] Phan, R. C.-W.: Cryptanalysis of a New Ultra lightweight RFID Authentication Protocol – SASI. IEEE Transactions on Dependable and Secure Computing, vol. 6, no. 4, pp. 316-320, Oct-Dec 2009.

[20] Van Deursen, T., Mauw, S., Radomirovic, S.: Untraceability of RFID Protocols. J.A. Onieva et al. (Eds.): WISTP 2008, LNCS 5019, pp. 1–15, 2008.

[21] Habibi, M. H., Gardeshi, M., Alagheband, M.R.: Cryptanalysis of two mutual authentication protocols for low-cost RFID. International Journal of Distributed and Parallel systems, vol. 2, no. 1, pp. 103-114.

[22] Yeh, T.-C., Wang, Y.-J., Kuo, T.-C., Wang, S.-S.: Securing RFID systems conforming to EPC Class-1 Generation-2 standard. Expert Systems with Applications 37 (2010) 7678–7683.

[23] Avoine, G.: Adversarial model for radio frequency identification. Cryptology ePrint Archive, report 2005/049. http://eprint.iacr.org/2005/049.

[24] Juels, A., and Weis, S.A.: Defining strong privacy for RFID. In Proceedings of PerCom '07 (2007) 342–347, http://eprint.iacr.org/2006/137

[25] Vaudenay, S.: On Privacy Models for RFID. K. Kurosawa (Ed.): ASIACRYPT 2007, LNCS 4833, pp. 68–87, 2007.

[26] Ouafi, K., and Phan, R.C.-W.: Privacy of recent RFID authentication protocols. L. Chen, Y. Mu, and W. Susilo (Eds.): ISPEC 2008, LNCS 4991, pp. 263–277, 2008.