

This file has been cleaned of potential threats.

If you confirm that the file is coming from a trusted source, you can send the following SHA-256 hash value to your admin for the original file.

5ee6ef0b8b6501d7ce67c94cca536e662b3cc7b2812cd57d4bf2349b5d583939

To view the reconstructed contents, please SCROLL DOWN to next page.



Cryptanalysis and Improvement on a New RFID Mutual Authentication Protocol Compatible with EPC Standard

Mohammad Hassan Habibi

Faculty of Electrical Engineering, I.H University
Tehran, Iran
mohamad.h.habibi@gmail.com

Mahmud Gardeshi

Faculty of Electrical Engineering, I.H University
Tehran, Iran
m.gardeshi2000@yahoo.com

Abstract—The developing of RFID systems in sensitive applications like e-passport, e-health, credit cards, and personal devices, makes it necessary to consider the related security and privacy issues in precise. Among other security characteristic of an RFID authentication protocol, the untraceability and synchronization are the most important attributes. The former is strongly related to the privacy of tags and their holders, while the later has a significant role in the security and availability parameters. In this paper we investigate the recent proposed RFID authentication protocol by Yeh, Wang, Kuo, and Wang in terms of privacy and security. At first, 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 *untraceability* and *backward untraceability* attributes. We also will propose our revision to safeguard the Yeh et al.'s protocol against the cited attacks. (Abstract)

Keywords- RFID, EPC C-1 G-2 standard, Security analysis, Desynchronization Attacks, Untraceability (key words)

I. INTRODUCTION

Radio frequency Identification (RFID) systems have been developed rapidly in the last decade. They are used to identify objects wirelessly in many applications like public transportation passes [1], supply chain management [2], e-passport [3] etc. The history of using RFID technology is referred to the *World War II* during which a system called "IFF" (Identification, Friend or Foe) was applied. The IFF system allowed the British military to determine if incoming aircraft were friendly or hostile. However, with regard to the cost of developing RFID systems, the use of them was less popular until low-cost RFID tags were produced in the last decade.

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.

II. RELATED WORK

In this section we briefly study some works conforming to EPC C-1 G-2 standard which had aimed to provide secure communication for RFID systems. Since EPC tags are restricted by EPC C-1 G-2 standard, each of these protocols is vulnerable to some security flaws.

Juels [23] showed that cloning and counterfeiting attacks are applied simply on EPC tags. He proposed an unclonable authentication protocol to solve these problems. However, Duc et al. [15] presented some weaknesses related to privacy and information leakage on Juels scheme.

In [24] Karthikeyan and Nesterenko suggested a security protocol without complex cryptographic primitives. Only XOR and matrix operations were used in their scheme. Chien and Chen [10] showed that this protocol was vulnerable to replay attack and did not assure the *untraceability* property.

A mutual authentication protocol conforming to EPC C-1 G-2 standard was proposed by Chien and Chen [10]. They had used simple XOR, CRC and PRNG in their scheme. In [10] each tag needs to keep an EPC code and two secret keys K_i , P_i . Secret key K_i is used to tag authentication and secret key P_i is used to reader authentication. Both K_i and P_i are updated in each round whereas EPC code is fix. For each tag the values K_{old} , P_{old} , K_{new} , P_{new} , EPC and DATA are stored in the database. The protocol is initiated with sending a random number N_R by the reader. The tag replies with $M1, N_T$ where $M1 = CRC(EPC \parallel N_R \parallel N_T) \oplus K_i$. Receiving the tag's response, the database searches for finding the correct tag and its corresponding information ($\{K_{old}, P_{old}\}$ or $\{K_{new}, P_{new}\}$). Then the database computes $M2 = CRC(EPC \parallel N_T) \oplus P_x$ ($x = old$ or new) and sends tag $M2$. At this point the database updates its secret keys as following: $K_{old} = K_{new}$, $P_{old} = P_{new}$, $K_{new} = PRNG(K_{new})$ and $P_{new} = PRNG(P_{new})$. The tag receives $M2$ and checks whether $M2 \oplus P_i = CRC(EPC \parallel N_T)$. If it is satisfied, the tag authenticates the database and updates K_i and P_i the same as the database.

Lopez et al. [18] showed some weaknesses on Chien and Chen's protocol including tag and reader impersonation and desynchronization attack. They also showed this protocol did not guarantee forward security and it was vulnerable to tracing attack. Han and Kwon [16] also presented a desynchronization attack and two tag impersonation attacks on Chien and Chen's

protocol in the new methods. These attacks were mainly based on weak secure properties of CRC.

III. REVIEW SRP

A. Notations

We use the notations the same as the notations were used in the original paper [22].

\mathcal{A} : malicious adversary

EPC_S : 16-bit string which is built by XOR ing six 16-bit blocks of EPC code

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

DATA: The corresponding record for the tag kept in the database

H (.): Hash function

K_i : The authentication key stored in the tag for the database to authenticate the tag 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_i : The access key stored in the tag for the tag to authenticate the database at the $(i+1)$ th authentication phase

P_{old} : The old access key stored in the database

P_{new} : The new access key stored in the database

\mathcal{R} : the legitimate reader

\mathcal{T} : the legitimate tag

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

N_Y : The random number generated by device Y ($Y = R$ or T)

$\gamma_i^{T_i}$: The parameter γ related to the tag T_i at time t_j

\oplus : bitwise XOR

B. 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 .

C. The $(i+1)$ th Authentication Round

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

The reader generates number N_R at random and sends it to the tag.

Receiving N_R , first the tag generates random number N_T , then it computes:

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

$$D = N_T \oplus K_i$$

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

Now the tag forwards $(C_i, M1, D, E)$ to the reader.

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



Receives (C_i, MI, 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 $(EPC_s \oplus N_R)$, $I_{old} = MI \oplus K_{old}$ and $I_{new} = MI \oplus K_{new}$ 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 MI or not.

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

- c) Computes $M2 = \text{PRNG}(EPC_s \oplus N_T) \oplus P_X$ and $Info = (DATA \oplus RID)$, and sends them to the reader.
- d) If $X = new$, it updates the stored values as follows: $K_{old} = K_{new}$, $K_{new} = \text{PRNG}(K_{new})$, $P_{old} = P_{new}$, $P_{new} = \text{PRNG}(P_{new})$, $C_{old} = C_{new}$, $C_{new} = \text{PRNG}(N_T \oplus N_R)$. But if $X = old$, it just updates C_{new} as $C_{new} = \text{PRNG}(N_T \oplus N_R)$.

IV. 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*.

A. 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. Because the construction of the underlying PRNG is public, the adversary knows this construction.

1. The adversary starts a session with the target tag T_i in the round $(i+1)$ by sending random number N_{R1} and 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 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, T_i does not update its secret key K_i for the second session. Hence $M1_1$ and $M1_2$ are constructed as follows:
 $M1_1 = \text{PRNG}(EPC_s \oplus N_{R1}) \oplus K_i$,
 $M1_2 = \text{PRNG}(EPC_s \oplus N_{R2}) \oplus K_i$.
3. \mathcal{A} omits K_i by XORing $M1_1$ and $M1_2$:
 $M1_1 \oplus M1_2 = \text{PRNG}(EPC_s \oplus N_{R1}) \oplus K_i \oplus \text{PRNG}(EPC_s \oplus N_{R2}) \oplus K_i = \text{PRNG}(EPC_s \oplus N_{R1}) \oplus \text{PRNG}(EPC_s \oplus N_{R2}) = \beta$.
Where β is a 16-bit string as a result of $M1_1 \oplus M1_2$.
4. Let $L = \{l_1, l_2, \dots, l_{2^{16}}\}$ 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 $l_i \in L$

$\alpha = \text{PRNG}(l_i \oplus N_{R1}) \oplus \text{PRNG}(l_i \oplus N_{R2})$, 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.

B. 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 R and the target tag 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 $K_{i+1} = \text{PRNG}(K_i)$. 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 $C_{i+1} = \text{PRNG}(N_{T3} \oplus N_{R3})$.

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

C. 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. 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 $PRNG(EPC_s \oplus N_{T6})$. Then \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

D. 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.

1) Privacy Model

There are privacy models for the evaluation of RFID protocols [6, 25, 26, 27, 28]. We analyze SRP protocol based on Ouafi and Phan model [28] which is based on [26] 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** ($i, \mathcal{T}_0, \mathcal{T}_1$) 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 $\mathcal{T}_b \in \{\mathcal{T}_0, \mathcal{T}_1\}$. 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}_0 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** ($i, \mathcal{T}_0, \mathcal{T}_1$) 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} 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 $\text{Adv}_A^{\text{UPriv}}(k)$ where k is the security parameter.

$$\text{Adv}_A^{\text{UPriv}}(k) = |\text{pr}(b = b') - \text{pr}(\text{random flip coin})| = |\text{pr}(b' = b) - \frac{1}{2}| \quad \text{where } 0 \leq \text{Adv}_A^{\text{UPriv}}(k) \leq \frac{1}{2}.$$

In [6], 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 $t' < t$ ".

2) Backward Traceability Attack

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 thus 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** ($i, \mathcal{T}_0, \mathcal{T}_1$) 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** ($\mathcal{R}, \mathcal{T}_b, i$) 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' :

He computes $\text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_b}) \oplus M1_{i-1}^{T_b} = \theta$ where θ is a 16-bit string.

\mathcal{A} utilizes the following simple decision rule:

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

Thus we have:

$$\text{Adv}_A^{\text{UPriv}}(k) = |\text{pr}(b' = b) - \text{pr}(\text{random flip coin})| = |\text{pr}(b' = b) - \frac{1}{2}| = |1 - \frac{1}{2}| = \frac{1}{2} \gg \epsilon$$

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

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

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow M1_{i-1}^{T_b} = M1_{i-1}^{T_0} = \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_0}) \oplus K_{i-1}^{T_0} \quad (2)$$

$$(1), (2) \Rightarrow \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_b}) \oplus M1_{i-1}^{T_b} = \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_0}) \oplus M1_{i-1}^{T_0} = \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_0}) \oplus \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i-1}^{T_0}) \oplus K_{i-1}^{T_0} = K_{i-1}^{T_0} = \theta \quad (3)$$

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow D_{i-1}^{T_b} \oplus E_{i-1}^{T_b} = D_{i-1}^{T_0} \oplus E_{i-1}^{T_0} = N_{T,i-1}^{T_0} \oplus K_{i-1}^{T_0} \oplus N_{T,i-1}^{T_0} \oplus \text{PRNG}(C_{i-1}^{T_0} \oplus K_{i-1}^{T_0}) = K_{i-1}^{T_0} \oplus \text{PRNG}(C_{i-1}^{T_0} \oplus K_{i-1}^{T_0}) = \theta \oplus \text{PRNG}(C_{i-1}^{T_0} \oplus \theta) = \theta \oplus \text{PRNG}(C_{i-1}^{T_b} \oplus \theta) \quad (4)$$

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 $N_{R,i}$ 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 $N_{R,i}$ 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:

$$\text{Adv}_A^{\text{UPriv}}(k) = |\text{pr}(b' = b) - \text{pr}(\text{random flip coin})| = |\text{pr}(b' = b) - \frac{1}{2}| = |1 - \frac{1}{2}| = \frac{1}{2}$$

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

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

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

Note that 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:

$$\text{If } \mathcal{T}_b = \mathcal{T}_0 \Rightarrow M1_i^{T_b} = \text{PRNG}(EPC_{s,i}^{T_b} \oplus N_{R,i}) \oplus K_i^{T_b} = \text{PRNG}(EPC_{s,i}^{T_0} \oplus N_{R,i}) \oplus K_i^{T_0} = M1_i^{T_0} \quad (7)$$

V. REVISED PROTOCOL

In order to eliminate the mentioned vulnerabilities in A to D subsections, we can modify the message MI as: $MI = \text{PRNG}(EPC_s \oplus N_R \oplus P_i) \oplus K_i$. Although the cited vulnerabilities are fixed by the above modification, the traceability problem still will be unsolved. Hence, we need to construct the message MI as following: $MI = \text{PRNG}(EPC_s \oplus N_R \oplus N_T) \oplus K_i$ to provide a secure protocol against all cited attacks.

A. 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 A subsection happened successfully. We have added the random and fresh value N_T in construction of MI 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} . Thus it is a noticeable improvement on SRP security.

VI. 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 M_1 in a new way..

REFERENCES

- [1] Transport for London, Oyster card, <http://www.oystercard.co.uk>
- [2] "Michelin Embeds RFID Tags in Tires". RFIDJournal. <http://www.rfidjournal.com/article/articleview/269/1/1/>. Accessed 17 Jan 2003
- [3] D. Carluccio, K. Lemke-Rust, C. Paar and A.-R. Sadeghi, "E-Passport: The global traceability or how to feel like an UPS package", In J. K. Lee, O. Yi, and M. Yung (eds) Proc. WISA 2007, Lecture Notes in Computer Science, Vol.4298 (Springer, Berlin), 2007, 391–404.
- [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] C.H. Lim, and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer", In Proceedings of ICICS '06, LNCS 4307 (2006) 1–20
- [7] T. van Deursen and S. Radomirovic, "Attacks on RFID protocols", Cryptology ePrint Archive, Report 2008/310, 2008. <<http://eprint.iacr.org/>>.
- [8] K. Ouafi and R.C.-W. Phan, "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] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda, "Vulnerability analysis of RFID protocols for tag ownership transfer", Computer Networks 54 (2010) 1502–1508.
- [10] H. Y Chien, and C. H. Chen, (2007), "Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards", Computer Standards & Interfaces, 29(2), 254–259. doi:10.1016/j.csi.2006.04.004
- [11] D. M. Konidala and K. Kim, "Rfid tag-reader mutual authentication scheme utilizing tag's access password" Auto-ID Labs White Paper WP-HARDWARE-033, 2007.
- [12] L. Kulseng, Z. Yu, Y. Wei and Y. Guan : "Lightweight mutual authentication and ownership transfer for RFID Systems", In Proceedings of IEEE INFOCOM 2010, San Diego, CA, March (2010) 1-5
- [13] H.-Y. Chien, "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] B. Song and C.J. Mitchell, "RFID authentication protocol for low-cost tags", In Wiscac 2008, pages 140-147
- [15] D. N. Duc, J. Park, H. Lee, and K. Kwangjo, "Enhancing security of epcglobal Gen-2 RFID tag against traceability and cloning", In Proc. of Symposium on Cryptography and Information Security, 2006.
- [16] D. Han, D. Kwon, "Vulnerability of an RFID authentication protocol conforming to EPC Class-1 Generation-2 Standards", Computer Standards & Interfaces 31 (2009) 648–65
- [17] M. H. Habibi, M. Gardeshi, M. Alaghband, "Attacks and improvements to a new RFID mutual Authentication protocol", Third Workshop on RFID Security: RFIDsec Asia 2011, China, 2011.
- [18] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, " Practical attacks on a mutual authentication scheme under the EPC Class-1 Generation-2 standard", Computer Communications 32 (2009) 1185–1193
- [19] M. H. Habibi, M. Gardeshi, M. Alaghband, "Cryptanalysis of two mutual authentication protocols for low-cost RFID", International Journal of Distributed and Parallel systems, Volume 1, Number 3, (2011)
- [20] R. C.-W. Phan, "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.
- [21] M. H. Habibi, M. Gardeshi, M. Alaghband, "Security analysis of an RFID mutual authentication protocols for RFID systems", In proceedings of IEEE International Conference on Intelligent Information Networks (ICIIN 2011), 2011, Dubai, UAE.
- [22] T.-C. Yeh, Y.-J. Wang, T.-C. Kuo, S.-S. Wang, "Securing RFID systems conforming to EPC Class-1 Generation-2 standard", Expert Systems with Applications 37 (2010) 7678–7683
- [23] A. Juels, "Minimalist cryptography for low-cost RFID tags" In Proc. Of SCN'04, volume 3352 of LNCS, pp. 149–164. Springer-Verlag, 2004.
- [24] S. Karthikeyan, M. Nesterenko, "RFID security without extensive cryptography", Proc. of SASN '05, ACM (2005) 63–67
- [25] G. Avoine, "Adversarial model for radio frequency identification", Cryptology ePrint Archive, report 2005/049. <http://eprint.iacr.org/2005/049>
- [26] A. Juels, and S.A. Weis, "Defining strong privacy for RFID", In Proceedings of PerCom '07 (2007) 342–347, <http://eprint.iacr.org/2006/137>
- [27] S. Vaudenay, "On Privacy Models for RFID", K. Kurosawa (Ed.): ASIACRYPT 2007, LNCS 4833, pp. 68–87, 2007.
- [28] K. Ouafi and R.C.-W. Phan, "Privacy of recent RFID authentication protocols", L. Chen, Y. Mu, and W. Susilo (Eds.): ISPEC 2008, LNCS 4991, pp. 263–277, 2008.