Security Analysis of an Ultra-lightweight RFID Authentication Protocol for M-commerce

Seyed Farhad Aghili and Hamid Mala

Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

{sf.aghili@eng, h.mala@eng}.ui.ac.ir

Abstract. Over the last few years, more people perform their social activities on mobile devices, such as mobile payment or mobile wallet. Mobile commerce (m-commerce) refers to manipulating electronic commerce (e-commerce) by using mobile devices and wireless networks. Radio frequency identification (RFID) is a technology which can be employed to complete payment functions on m-commerce. As an RFID subsystem is applied in m-commerce and supply chains, the related security concerns is very important. Recently, Fan *et al.* have proposed an ultra-lightweight RFID authentication scheme for m-commerce (*ULRAS*) and claimed that their protocol is enough efficient, and provides a high level of security. In this paper, we show that their protocol is vulnerable to secret disclosure and reader impersonation attacks. Finally, we improve the Fan *et al.* protocol to present a new one, which is resistant to the mentioned attacks presented in this paper and the other known attacks in the context of RFID authentication. Our proposed improvement does not impose any additional workload on the RFID tag.

keywords: Mobile commerce, RFID, Ultra-lightweight, Secret disclosure, Impersonation.

1 Introduction

In the last few years, many business areas are getting more and more electronic and the need for electronic commerce (e-commerce) is increasing rapidly. In addition, mobile communication is increasing dramatically such that more than half of the population of the world have mobile phones and drivers of mobile commerce (m-commerce) which enables them to manipulate their e-commerce affairs. M-commerce has many applications, such as banking and financial services, mobile enterprise applications, ubiquitous computing, mobile shopping, mobile marketing and advertising, mobile payment and, so on [27]. One of the most important technologies of mobile devices is RFID (Radio Frequency Identification) which is employed to enable mobile

payments wirelessly. However, in RFID systems reader and tag use the radio channel for transferring the important information, which is insecure. To resolve the above problems, many solutions have been proposed to secure RFID systems [4, 5, 7, 9, 16, 19, 23, 25, 29, 30], but most of proposed protocols still suffer from various security vulnerabilities [1, 23, 29, 16]. As deal with very cheap barcodes, low cost RFID tags must be used for m-commerce. So, only ultra-lightweight RFID schemes can be compatible with these kinds of tags which consume less computing and storage resources [28].

In order to overcome these problems, many protocols have been proposed for authenticating low cost RFID tags in RFID systems. For example, MAP-family (EMAP, M2AP, LMP$^+$ and etc.) [21, 22, 18] based on bitwise operations like AND, XOR and OR and the HB-family (HB, HB$^+$,HB^{++} and etc.) [12, 14, 3] by employing matrix multiplication and some XORs are some of the lightweight authentication protocols proposed in the literatures. However, these two models have several limitations, weaknesses and vulnerabilities [11, 20, 13, 26, 31]. Later, in [17], Kulseng *et al.* proposed a lightweight solution to mutual authentication for RFID systems by using Linear Feedback Shift Registers (LFSRs) and Physically Unclonable Functions (PUFs) which are lightweight operations. However, Kardas [15] showed that their protocol is not resistant against message injection attack, and has several vulnerabilities.

In the recent decade, the first ultra-lightweight protocol called *SASI* was proposed in [8] which is based on bit-wise functions such as XOR and rotation operations. However, this protocol has several vulnerabilities proposed in [6]. In 2009, the authors in [24] proposed another ultra-lightweight protocol called *Gossamer* to improve the security weaknesses in ultra-lightweight protocols. Later, in [2] it was shown that *Gossamer* protocol is also vulnerable to several attacks.

Recently, an ultra-lightweight RFID authentication protocol has been proposed by Fan *et al.* with the claim of being fit for m-commerce [10]. In this protocol, the authors employed simple operations such as bitwise XOR ($\oplus$) and addition modulo 2^L ($+$) and also shift operation (called *RR* method) they also claimed that their protocol is secure and efficient enough. In this paper, we show that their protocol is vulnerable to secret disclosure and reader impersonation attacks.

Paper organization: Preliminaries and notations used in this paper are mentioned in Section 2. A brief description of Fan *et al.* scheme [10] is provided in Section 3. We analyze the security of Fan *et al.* protocol in Section 4, and propose several attacks against this protocol. In Section 5, we present our improvement. Security and efficiency of the improved protocol are presented in Sections 6, respectively. Finally, the paper is concluded in Section 7.

2 Notations and Preliminaries

Notations used throughout this paper are depicted in Tabel 1. The Fan *et al.* scheme and our improvement use simple operations such as bitwise XOR ($\oplus$) and addition modulo 2^L ($+$) and also shift operation ($Rot(X,Y)$). The operation $Rot(X,Y)$ is defined as the circular shift on the value $X \oplus Y$ by $(Y \ mod \ L)$ bits to the left for a given value of L, where L is the length of parameters X and Y(called *RR* method) [10].

Table 1. Notations used in this paper

T_R	Random time stamp generated by the reader
T_T	The last time stamp stored in the tag
R_T	Random number generated by the tag
IDS	The index number of the tag
IDS_{old}	The index number used in the last time
IDS_{new}	The index number used this time
K	Shared key between the tag and its owner
K_{old}	The key of the tag used in the last time
K_{new}	The key of the tag used this time
i_{sub}	The random index number, $i_{sub} \in \{1,2,3,4\}$
$K(i_{sub})$	The old sub-key indexed by i_{sub}
$\oplus$	Exclusive OR operation
$+$	Addition modulo 2^L
$\|\|$	Concatenation operation
$Rot(X,Y)$	The rotation of X according to Y

3 Review of the $ULRAS$ Protocol

Recently Fan *et al.* proposed an ultra-lightweight authentication protocol (called $ULRAS$) and claimed that their protocol has high efficiency and strong security [10]. The $ULRAS$ protocol is depicted in Fig. 1 and discribed as below:

3.1 The Initialization Phase

In this phase of the protocol, the server S stores the tag's records $(ID,(IDS_{old},K_{old}),(IDS_{new},K_{new}))$ that are unique for each tag T, and each tag stores the tuple (IDS,ID,K,T_T).

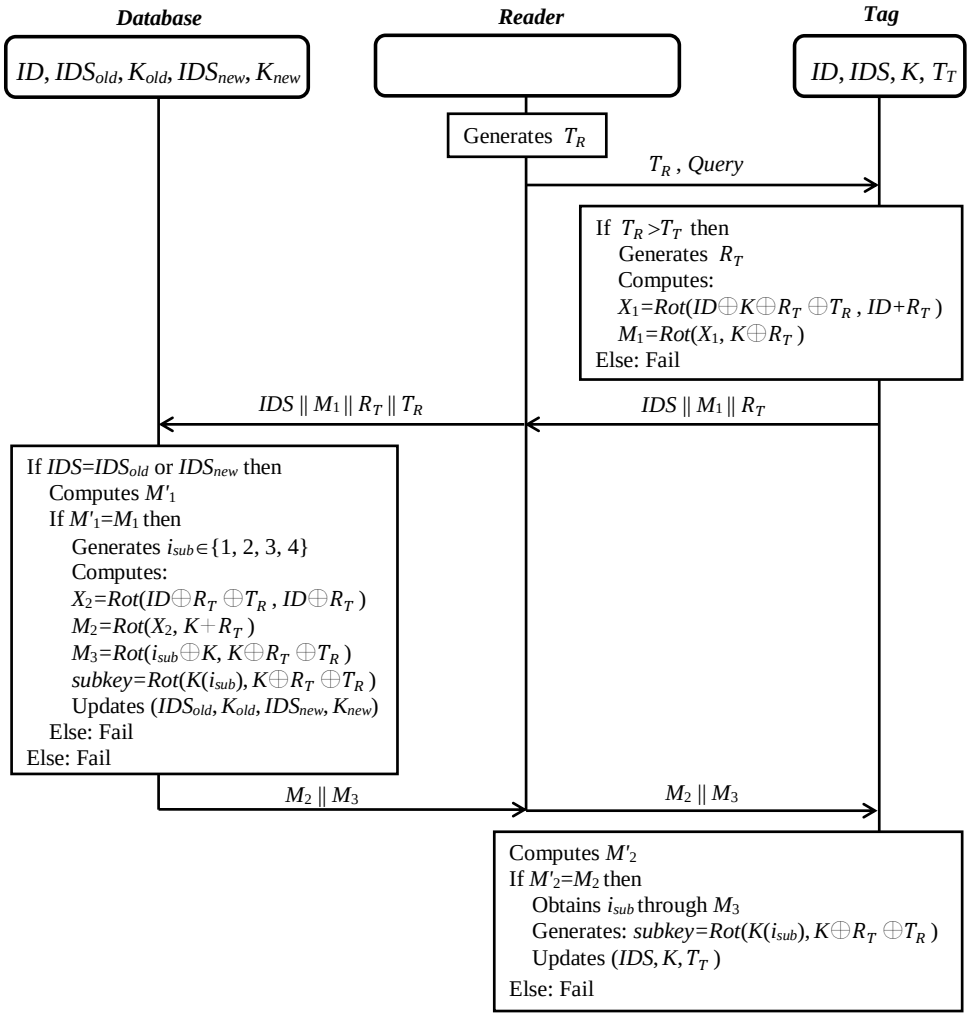

Fig. 1. The *ULRAS* protocol [10]

3.2 The Authentication Phase

The authentication phase workes as follows:

- *Step 1.* The reader R generates a random time stamp T_R which is greater than T_T and sends it to the T along with Query.
- *Step 2.* Upon receiving the Query, T checks whether T_R is greater than T_T. If true, T generates a random number R_T and computes $X_1 = Rot(ID \oplus K \oplus R_T \oplus T_R, ID + R_T)$ and uses X_1 to compute $M_1 = Rot(X_1, K \oplus R_T)$, then T sends the tuple (IDS, M_1, R_T) to R; otherwise, the protocol ends with failure.
- *Step 3.* After receiving the tuple (IDS, M_1, R_T), R sends the message $IDS \| M_1 \| R_T \| T_R$ to S.
- *Step 4.* Upon receiving the tuple (IDS, M_1, R_T, T_R), S checks whether IDS exists in its database. If S finds a match for IDS, it executes Step 5; otherwise, it ends the protocol with failure.
- *Step 5.* S uses the found $IDS = IDS_{old}$ or $IDS = IDS_{new}$ and computes $X_1' = Rot(ID \oplus K \oplus R_T \oplus T_R, ID + R_T)$ and then uses X_1' to compute $M_1' = Rot(X_1', K \oplus R_T)$, then S judges whether $M_1 = M_1'$. If true, S authenticates T and computes $X_2 = Rot(ID \oplus R_T \oplus T_R, ID \oplus R_T)$ and uses X_2 to compute $M_2 = Rot(X_2, K + R_T)$, then it generates the random index number $i_{sub} \in \{1,2,3,4\}$ to calculate $M_3 = Rot(i_{sub} \oplus K, K \oplus R_T \oplus T_R)$. Finally, the tuple (M_2, M_3) is sent to T through R.
- *Step 6.* Once T receives the messages M_2 and M_3, it computes $X_2' = Rot(ID \oplus R_T \oplus T_R, ID \oplus R_T)$ and uses X_2' to compute $M_2' = Rot(X_2', K + R_T)$, then it checks whether $M_2 = M_2'$. If yes, T obtains i_{sub} through M_3 and then generates $subkey = Rot(K(i_{sub}), K \oplus R_T \oplus T_R)$ and updates K and IDS, where $IDS_{new} = Rot(IDS \oplus R_T, K \oplus R_T \oplus T_R)$ and K_{new} is updated by replacing the $K(i_{sub})$ by the $subkey$. Finally, T rewrites T_T by T_R.

  At the same time, S generates a new sub-key ($subkey = Rot(K(i_{sub}), K \oplus R_T \oplus T_R)$ and if $IDS = IDS_{old}$, updates K_{new} and IDS_{new}, where $IDS_{new} = Rot(IDS \oplus R_T, K \oplus R_T \oplus T_R)$) and K_{new} is updated by replacing the $K(i_{sub})$ by the $subkey$. Otherwise, if $IDS = IDS_{new}$, then S rewrites IDS_{old} by IDS_{new} and K_{old} by K_{new} and then computes IDS_{new} and K_{new} with the same operation as described previously.

4 Security Analysis of the *ULRAS* Protocol

In this section, we show that the *ULRAS* protocol is volnerable to secret disclosure and reader impersonation attacks.

4.1 Secret Disclosure Attack

In this section, we show that it is possible to disclose the secret parameter K in $ULRAS$ protocol. The main two definitions in this attack are:

- $Rotl(Z,W)$ is defined as the circular shift of the value Z by W bits to the left;
- $Rotr(Z,W)$ is defined as the circular shift of the value Z by W bits to the right.

Online Phase:
Eavesdrop the first run of the mutual authentication protocol and store messages $M_2 = Rot(X_2, y)$ and $M_3 = Rot(i_{sub} \oplus K, p)$, where $y = K + R_T$, $X_2 = T_R \lll j$, $j = (ID \oplus R_T)\ mod\ L$, $p = K \oplus R_T \oplus T_R$ and $i = y\ mod\ L$.

Offline Phase:

```
for j = 0 : L − 1
  X2' ← TR ⋘ j
  for i = 0 : L − 1
    b ← M2 ⋙ i
    y' ← X2' ⊕ b
    if y' mod L = i then
      if y' − RT ≥ 0 then
        K' ← y' − RT
      else if y' − RT < 0 then
        K' ← 2^L + y' − RT
      end if
    end if
    for i'sub = 1 : 4
      M3' ← (i'sub ⊕ RT ⊕ TR) ⋘ (K' ⊕ RT ⊕ TR) mod L
      if M3' = M3 then
        K'' ← K'
        output (j, i'sub, K'')
      end if
    end
  end
end
```

Decision Phase:
With the success probability of "$\frac{1}{L}$", $K' = K$ and with the success probability of "$\frac{1}{2L}$", $K'' = K$, so we have on average "$4L^2 \times (\frac{1}{2L} \times \frac{1}{L}) = 2$" values for K.

Algorithm 1: Secret disclosure attack against the $ULRAS$ protocol.

The attack consists of three phases, on-line phase, off-line phase and decision phase as follows:

On-line Phase: In an on-line phase of the attack, an adversary A does as follows:

– eavesdrops a session of the mutual authentication protocol and stores messages $M_2 = Rot(X_2, y)$ and $M_3 = Rot(i_{sub} \oplus K, p)$; where $y = K + R_T$, $X_2 = Rotl(T_R, j)$, $j = (ID \oplus R_T)\ mod\ L$, $p = K \oplus R_T \oplus T_R$ and $i = y\ mod\ L$.

Off-line Phase: In an off-line phase of the attack, the adversary A for $j = 0, \ldots, L-1$ does as follows:

– $X_2' \leftarrow Rotl(T_R, j)$

– for $i = 0, \ldots, L-1$ does as follows:

- $b \leftarrow Rotr(M_2, i)$;
- $y' \leftarrow X_2' \oplus b$.

– it checks whether $y'\ mod\ L = i$. For each matches, A concludes $y' = y$ and obtains $K' = y - R_T$ if $y - R_T \geq 0$ or $K' = 2^L + y - R_T$ if $y - R_T < 0$.

– for $i'_{sub} = 1, \ldots, 4$ does as follows:

- $M_3' \leftarrow Rotl[(i'_{sub} \oplus R_T \oplus T_R), (K' \oplus R_T \oplus T_R)\ mod\ L]$;
- it checks whether $M_3' = M_3$. For each matches, A concludes $i'_{sub} = i_{sub}$ and $K = K''$.

Decision Phase: In this phase of the attack, we show that $K' = K$ with the success probability of "$\frac{1}{L}$", and $K'' = K$ with the success probability of "$\frac{1}{2L}$", the computations are as follows:

$$\begin{aligned} Pr[K' = K] &= Pr[K' = K | i = y\ mod\ L, j = (ID \oplus R_T)\ mod\ L] \times \tfrac{1}{L^2} \\ &+ Pr[K' = K | i = y\ mod\ L, j \neq (ID \oplus R_T)\ mod\ L] \times \tfrac{L-1}{L^2} \\ &+ Pr[K' = K | i \neq y\ mod\ L, j = (ID \oplus R_T)\ mod\ L] \times \tfrac{L-1}{L^2} \\ &+ Pr[K' = K | i \neq y\ mod\ L, j \neq (ID \oplus R_T)\ mod\ L] \times \tfrac{(L-1)^2}{L^2} \\ &= \tfrac{1}{L^2} + \tfrac{1}{L}(\tfrac{L-1}{L^2} + \tfrac{L-1}{L^2} + \tfrac{(L-1)^2}{L^2}) \approx \tfrac{1}{L} \end{aligned}$$

and

$$\begin{aligned} Pr[K'' = K] &= Pr[M_3' = M_3] = Pr[M_3' = M_3 | i'_{sub} \neq i_{sub}, K' \neq K] \times (\tfrac{3}{4} \times \tfrac{L-1}{L}) \\ &+ Pr[M_3' = M_3 | i'_{sub} \neq i_{sub}, K' = K] \times (\tfrac{3}{4} \times \tfrac{1}{L}) \\ &+ Pr[M_3' = M_3 | i'_{sub} = i_{sub}, K' \neq K] \times (\tfrac{1}{4} \times \tfrac{L-1}{L}) \\ &+ Pr[M_3' = M_3 | i'_{sub} = i_{sub}, K' = K] \times (\tfrac{1}{4} \times \tfrac{1}{L}) \\ &= (\tfrac{1}{2})^L (\tfrac{3}{4} \times \tfrac{L-1}{L}) + (\tfrac{1}{2})^L (\tfrac{3}{4} \times \tfrac{1}{L}) + (\tfrac{1}{L})(\tfrac{1}{4} \times \tfrac{L-1}{L}) + (1)(\tfrac{1}{4} \times \tfrac{1}{L}) \approx \tfrac{1}{2L} \end{aligned}$$

Now, we have $4L^2$ tuples of (j, i, i_{sub}) that each of them satisfies $K' = K'' = K$ with the success probability of "$\frac{1}{2L^2}$". So, we have on average "$4L^2 \times (\frac{1}{2L} \times \frac{1}{L}) = 2$" values for K . Consequently, we obtain "2" values for each j and i_{sub} by employing the values of K.

Algorithm 1 briefly presents the secret parameter disclosure attack against the *ULRAS* protocol.

4.2 Reader Impersonation Attack

In this sub-section, we suppose that the adversary A has applied the secret disclosure attack presented in the previous sub-section and has obtained on average two tupleS (j, i'_{sub}, K''). So, A can use one of these tuples

Input: (j_1, i'_{sub1}, K''_1) and (j_2, i'_{sub2}, K''_2) % Obtained from the Algorithm 1
Input: (T_{Rold}, R_{Told}) % Transferred from R to T in the last successful session
Output: (j, i_{sub}, Key)
$(j, i_{sub}, Key) \leftarrow (j_1, i'_{sub1}, K''_1)$ % The first tuple of (j, i'_{sub}, K'') obtained from Algorithm 1
$subkey \leftarrow Rot(Key(i_{sub}), Key \oplus R_T \oplus T_R)$ % Used to update the value of Key
$Key(i_{sub}) \leftarrow subkey$ % The Key is updated
$K \leftarrow Key$
$X_2 \leftarrow Rotl(T_R, j)$
$M_2 \leftarrow Rot(X_2, K + R_T)$
$M_3 \leftarrow Rot(i_{sub} \oplus K, K \oplus R_T \oplus T_R)$
send (M_2, M_3) to T
wait until T responses % The response is either $accept$ or $fail$
$Resp \leftarrow \mathrm{T}$'s response
if $Resp = fail$ then
$(j, i_{sub}, Key) \leftarrow (j_2, i'_{sub2}, K''_2)$ % The secund tuple of (j, i'_{sub}, K'') obtained from Algorithm 1
Go to 2
end if
output (j, i_{sub}, Key)

Algorithm 2: Reader impersonation attack against the $ULRAS$ protocol.

to impersonates R to T according to the following steps:

- *Step1.* An adversary A eavesdrops values T_R and R_T transferred from R to T in the last successful session and uses K'' to generate $subkey = Rot(K''(i_{sub}), K'' \oplus R_T \oplus T_R)$ for calculating the updated K. Note that in $ULRAS$ protocol, T rewrites T_T by T_R when the protocol finishes successfully.
- *Step2.* A initiates a communication with T and transmits $T'_R > T_R$.
- *Step3.* T checks whether $T'_R > T_T = T_R$. Because the inequality holds, T generates a random number R_T and computes M_1, then T sends the tuple (IDS, M_1, R_T) to R which is impersonated by A.
- *Step 4.* A calculates $X_2 = Rotl(T_R, j)$ for computing $M_2 = Rot(X_2, K + R_T)$, and uses the updated K for computing $M_3 = Rot(i_{sub} \oplus K, K \oplus R_T \oplus T_R)$. Finally, the tuple (M_2, M_3) is sent to T.
- *Step 5.* Once T receives the messages M_2 and M_3, it computes $X'_2 = Rot(ID \oplus R_T \oplus T_R, ID \oplus R_T) = Rotl(T_R, (ID \oplus R_T) \ mod \ L)$ and uses X'_2 to compute $M'_2 = Rot(X'_2, K + R_T)$, then it checks whether $M_2 = M'_2$. If yes, T obtains i_{sub} through M_3 and then generates $subkey = Rot(K(i_{sub}), K \oplus R_T \oplus T_R)$ and updates

K and IDS, where $IDS_{new} = Rot(IDS \oplus R_T, K \oplus R_T \oplus T_R)$ and K_{new} is updated by replacing the $K(i_{sub})$ by the *subkey*. Finally, T rewrites T_T by T_R. If no, T returns fail to A.

If T returns fail to A, the attacker Arepeats the above attack by the second tuple of (j, i'_{sub}, K'').

Now, based on the above attack, the adversary can successfully impersonate R to $\mathrm{T}$with probability "1". In addition, the attacker can determine i_{sub} in $M_3 = Rot(i_{sub} \oplus K, K \oplus R_T \oplus T_R)$ message to necessitate $\mathrm{T}$to update K to a predetermined value. In other words the attacker can impersonate R to T permanently.

Note that by executing the reader impersonation attack presented in Algorithm 2, the attacker can easily find the correct tuple (j, i_{sub}, Key) between the two tuples (j_1, i'_{sub1}, K''_1) and (j_2, i'_{sub2}, K''_2) obtained in Algorithm 1.

5 The Improved Protocol

To improve the security flaws of the *ULRAS* protocol, we propose an ultra-lightweight authentication protocol that has strong security with the same efficiency in the tag side.

In our protocol, to cope with the presented attacks, we use both secret parameters of the tag (K and ID) in the messages of the reader and the tag. The proposed protocol is discribed as below (Fig. 2).

5.1 The Initialization Phase

In this phase of the proposed protocol, like the *ULRAS* protocol, the server S stores the tag's records $(ID, (IDS_{old}, K_{old}), (IDS_{new}, K_{new}))$ that are unique for each tag T, and each tag stores the tuple (IDS, ID, K, T_t).

5.2 The Authentication Phase

The authentication phase of the proposed protocol works as follows.

- *Step 1.* R generates a random time stamp T_R which is greater than T_T, and sends it to the T along with Query.
- *Step 2.* Upon receiving the Query, T checks whether T_R is greater than T_T. If true, T generates a random number R_T and computes $X_1 = Rot(ID \oplus K \oplus R_T \oplus T_R, ID + R_T)$ and uses X_1 to compute $M_1 = Rot(X_1, K \oplus R_T \oplus ID)$, then T sends the tuple (IDS, M_1, R_T) to R; otherwise, the protocol ends with failure.
- *Step 3.* After receiving the tuple (IDS, M_1, R_T), R sends the message $IDS \| M_1 \| R_T \| T_R$ to S.

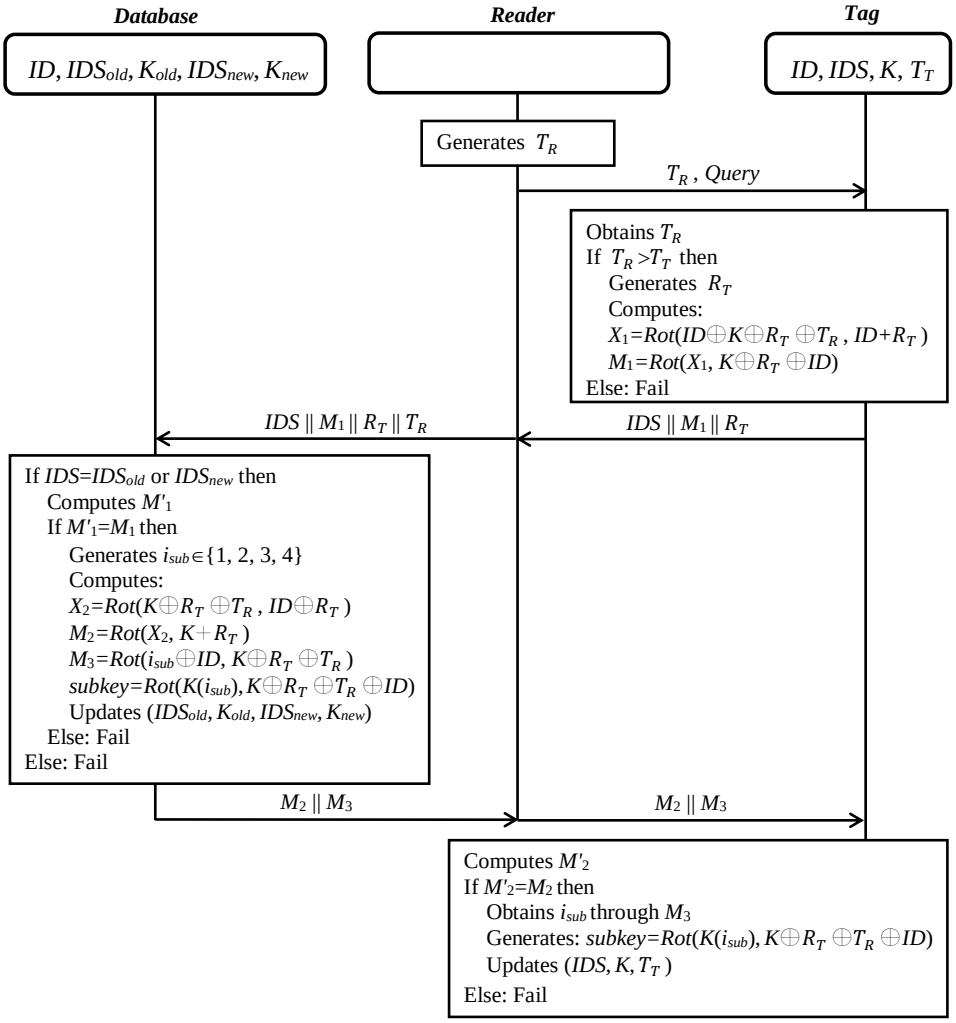

Fig. 2. Improved protocol

– *Step 4.* Upon receiving the tuple (IDS, M_1, R_T, T_R), S checks whether IDS exists in its database. If S can find a match for IDS, it executes Step 5; otherwise, it ends the protocol with failure.

– *Step 5.* S uses the found $IDS = IDS_{old}$ or $IDS = IDS_{new}$ and computes $X_1' = Rot(ID \oplus K \oplus R_T \oplus T_R, ID + R_T)$ and then uses X_1' to compute $M_1' = Rot(X_1', K \oplus R_T \oplus ID)$, then S judges whether $M_1 = M_1'$. If true, S authenticates T and computes $X_2 = Rot(K \oplus R_T \oplus T_R, ID \oplus R_T)$ and uses X_2 to compute $M_2 = Rot(X_2, K + R_T)$, then it generates the random index number $i_{sub} \in \{1,2,3,4\}$ to calculate $M_3 = Rot(i_{sub} \oplus ID, K \oplus R_T \oplus T_R)$. Finally, the tuple (M_2, M_3) is sent to T through R.

– *Step 6.* Once T receives the messages M_2 and M_3, it computes $X_2' = Rot(K \oplus R_T \oplus T_R, ID \oplus R_T)$ and uses X_2' to compute $M_2' = Rot(X_2', K + R_T)$, then it checks whether $M_2 = M_2'$. If yes, T obtains i_{sub} through M_3 and then generates $subkey = Rot(K(i_{sub}), K \oplus R_T \oplus T_R \oplus ID)$ and updates K and IDS, where $IDS_{new} = Rot(IDS \oplus R_T, K \oplus R_T \oplus T_R \oplus ID)$ and K_{new} is updated by replacing the $K(i_{sub})$ by the $subkey$. Finally, T rewrites T_T with T_R.

  At the same time, S generates a new sub-key ($subkey = Rot(K(i_{sub}), K \oplus R_T \oplus T_R \oplus ID)$ and if $IDS = IDS_{old}$, updates K_{new} and IDS_{new}, where $IDS_{new} = Rot(IDS \oplus R_T, K \oplus R_T \oplus T_R \oplus ID)$) and K_{new} is updated by replacing the $K(i_{sub})$ by the $subkey$. Otherwise, if $IDS = IDS_{new}$, then S rewrites IDS_{old} by IDS_{new} and K_{old} by K_{new} and then computes IDS_{new} and K_{new} with the same operation as described previously.

Table 2. Security features comparison between *SASI* , *Gossamer*, *ULRAS* protocol and our improved protocol.

	RSD	RRI	RTI	FBS	RR	RD	RT
SASI [8]	Yes	Yes	Yes	Yes	Yes	No	No
Gossamer [24]	Yes	Yes	Yes	Yes	Yes	No	Yes
ULRAS	No	No	Yes	Yes	Yes	Yes	Yes
Our improved protocol	Yes	Yes	Yes	Yes	Yes	Yes	Yes

RSD: Resistance against secret disclosure attack
RRI: Resistance against reader impersonation attack
RTI: Resistance against tag impersonation attack
FBS: Forward and backward security
RR: Resistance against replay attack
RD: Resistance against de-synchronization attack
RT: Resistance against traceability attack

6 Security Analysis of the Improved Protocol

In this section, we analyze the security of the proposed protocol and show that how we prohibit the security flaws of the *ULRAS* protocol.

6.1 Resistance to Secret Disclosure Attack

In Our proposed protocol, we use both secret parameters of the tag (K and ID) in the messages of the readers and the tags. So, an adversary cannot find any message with only one unknown parameter to execute presented secret disclosure attack.

6.2 Resistance to Traceability Attack

In our protocol, the tag computes all of the messages by employing the fresh random number (R_T). So, the tag's responses are neither constant nor predictable by an attacker and she cannot track the target tag.

6.3 Forward and Backward Security

In our proposed protocol, all of the messages are computed by irreversible function ($Rot(\cdot)$). So, we cannot find any message in which an adversary would be able to obtain the current and previous confidential information. Therefore, our proposal achieves forward security.

6.4 Resistance to De-synchronization Attack

If we assume an adversary can block the last message of the protocol (M_2, M_3) and cause that the tag do not update the tuple (IDS and K) to execute the de-synchronization attack, because the server stores the tuple ,($IDS_{old}, K_{old}, IDS_{new}, K_{new}$), it can use IDS_{old} and K_{old} to authenticate the tag. So, the adversary cannot render the tag to the de-synchronization state.

6.5 Resistance to Replay Attack

In our proposal, all of the messages are involved freshly generated random numbers (T_R and R_T). Therefore, an adversary cannot replay eavesdropped messages from previous sessions to cheat any entity involved in protocol. Hence, the improved protocol is robust against replay attack.

6.6 Resistance to Tag Impersonation Attack

Due to use of random number (T_R) computed by the reader in the tags messages and the authentication process in the server side, the adversary cannot send the expected answer to the reader. Therefore, the proposed protocol is immune against tag impersonation attack.

6.7 Resistance to Reader Impersonation Attack

In our protocol, the reader or the server computes M_2 and M_3 by employing the tag's random number (R_T). So, an adversary cannot replay eavesdropped messages (M_2, M_3) from previous sessions to deceive the tag.

Table 2 compares the serious security features of the *ULRAS* protocol and our improved protocol.

Table 3 depicts the performance comparison of *SASI* [8], *Gossamer* [24], *ULRAS* and our proposed protocol in tag side. The comparison shows that our proposal is as efficient as *ULRAS* protocol.

Table 3. Performance comparison between *SASI* , *Gossamer*, *ULRAS* and our proposed protocol.

Protocol	Tag computation
SASI	$\oplus, +, \vee, \wedge, Rot$
Gossamer	$\oplus, +, Rot^2, MIXBITS$
ULRAS	$T_R, \oplus, +, Rot^2$
Our proposal	$T_R, \oplus, +, Rot^2$
$\oplus$ is the bitwise XOR $+$ is the addition modulo 2^L $\vee$ is the bitwise OR $\wedge$ is the bitwise AND *Rot* is the shift operation Rot^2 is the twice shift operation	

7 Conclusion

In this paper we considered the security of an RFID mutual authentication protocol for m-commerce (*ULRAS*). In this protocol, authors aimed that computational overhead in their scheme is acceptable by using *RR* method and is efficient enough for low-cost RFID systems. They also claimed that their protocol provides the high level of security. However, we showed that an attacker can obtain the tag's key with high probability and can

also execute reader impersonation attack. Finally, we proposed the improved version which is secure and still suitable for low-cost RFID systems in m-commerce.

References

1. S. F. Aghili, N. Bagheri, P. Gauravaram, M. Safkhani, and S. K. Sanadhya. On the Security of Two RFID Mutual Authentication Protocols. In *RFIDSec*, Lecture Notes in Computer Science, pages 86–99. Springer, 2013.
2. Z. Bilal, A. Masood, and F. Kausar. Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol. In *Network-Based Information Systems, 2009. NBIS'09. International Conference on*, pages 260–267. IEEE, 2009.
3. J. Bringer, H. Chabanne, and E. Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06)*, pages 28–33. IEEE, 2006.
4. M. Burmester and B. de Medeiros. The Security of EPC Gen2 Compliant RFID Protocols. In S. M. Bellovin, R. Gennaro, A. D. Keromytis, and M. Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 490–506, 2008.
5. M. Burmester, B. de Medeiros, J. Munilla, and A. Peinado. Secure EPC Gen2 compliant Radio Frequency Identification. *IACR Cryptology ePrint Archive*, 2009:149, 2009.
6. T. Cao, E. Bertino, and H. Lei. Security analysis of the SASI protocol. *IEEE Transactions on Dependable and secure Computing*, 6(1):73–77, 2009.
7. C.-L. Chen and Y.-Y. Deng. 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.
8. 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.
9. H.-Y. Chien and C.-H. Chen. Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standards & Interfaces*, 29(2):254–259, 2007.
10. K. Fan, N. Ge, Y. Gong, H. Li, R. Su, and Y. Yang. An ultra-lightweight RFID authentication scheme for mobile commerce. *Peer-to-Peer Networking and Applications*, pages 1–9, 2016.
11. H. Gilbert, M. Robshaw, and H. Silvert. An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol. Technical report, Cryptology ePrint Archive, Report 2005/237, 2005, available at http://eprint. iacr. org/2005/237. pdf.
12. N. J. Hopper and M. Blum. Secure Human Identification Protocols. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 52–66. Springer, 2001.

13. S. Islam. Security analysis of LMAP using AVISPA. *International Journal of Security and Networks*, 9(1):30–39, 2014.

14. A. Juels and S. A. Weis. Authenticating Pervasive Devices with Human Protocols. In *Annual International Cryptology Conference*, pages 293–308. Springer, 2005.

15. S. Kardas, M. Akgün, M. S. Kiraz, and H. Demirci. Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems. In *Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on*, pages 20–25. IEEE, 2011.

16. S. Karthikeyan and M. Nesterenko. RFID security without extensive cryptography. In V. Atluri, P. Ning, and W. Du, editors, *SASN*, pages 63–67. ACM, 2005. Proceedings of the 3rd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2005, Alexandria, VA, USA, November 7, 2005.

17. L. Kulseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight Mutual Authentication and Ownership Transfer for RFID Systems. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE, 2010.

18. T. Li. Employing Lightweight Primitives on Low-Cost RFID Tags for Authentication. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE, 2008.

19. I.-C. Lin, R.-K. Luo, and S.-C. Tsao. An Efficient Mutual Authentication Protocol for RFID Systems. In G. Yu, M. Köppen, S.-M. Chen, and X. Niu, editors, *HIS (3)*, pages 41–45. IEEE Computer Society, 2009. 9th International Conference on Hybrid Intelligent Systems (HIS 2009), August 12-14, 2009, Shenyang, China.

20. K. Ouafi, R. Overbeck, and S. Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 108–124. Springer, 2008.

21. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual-Authentication Protocol for Low-Cost RFID Tags. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 352–361. Springer, 2006.

22. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. M^2AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags. In *International Conference on Ubiquitous Intelligence and Computing*, pages 912–923. Springer, 2006.

23. P. Peris-Lopez, J. C. Hernandez-Castro, J. E. Tapiador, and J. C. A. van der Lubbe. Cryptanalysis of an EPC Class-1 Generation-2 standard compliant authentication protocol. *Eng. Appl. of AI*, 24(6):1061–1069, 2011.

24. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Tapiador, and A. Ribagorda. Advances in ultra-lightweight cryptography for low-cost RFID tags: Gossamer protocol. In *International Workshop on Information Security Applications*, pages 56–68. Springer, 2008.

25. P. Peris-Lopez, T. Li, and J. C. Hernandez-Castro. Lightweight Props on the Weak Security of EPC Class-1 Generation-2 Standard. *IEICE Transactions*, 93-D(3):518–527, 2010.

26. M. Safkhani, N. Bagheri, M. Naderi, and S. K. Sanadhya. Security analysis of LMAP^{++}, an RFID authentication protocol. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 689–694. IEEE, 2011.
27. E. Turban, J. Whiteside, D. King, and J. Outland. Mobile commerce and the internet of things. In *Introduction to Electronic Commerce and Social Commerce*, pages 167–199. Springer, 2017.
28. C.-H. Wei, M.-S. Hwang, and A. Y.-H. Chin. A secure privacy and authentication protocol for passive RFID tags. *International Journal of Mobile Communications*, 15(3):266–277, 2017.
29. T.-C. Yeh, Y.-J. Wang, T.-C. Kuo, and S.-S. Wang. Securing RFID systems conforming to EPC Class 1 Generation 2 standard. *Expert Syst. Appl.*, 37(12):7678–7683, 2010.
30. E.-J. Yoon. Improvement of the securing RFID systems conforming to EPC Class 1 Generation 2 standard. *Expert Syst. Appl.*, 39(1):1589–1594, 2012.
31. F. Zeng, H. Mu, and X. Wen. An Improved LMAP++ Protocol Combined with Low-Cost and Privacy Protection. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, pages 847–853. Springer, 2014.