An Enhanced Mutual Key Agreement Protocol for Mobile RFID-enabled Devices

Kambombo Mtoga[1] and Eun-Jun Yoon[2]

Abstract – Mobile RFID is a new application that uses a mobile phone as an RFID reader with wireless technology and provides a new valuable service to users by integrating RFID and ubiquitous sensor network infrastructures with mobile communication and wireless Internet. Whereas the mobile RFID system has many advantages, privacy violation problems on the reader side are very concerning to individuals and researchers. Unlike in regular RFID environments, where the communication channel between the server and reader is assumed to be secure, the communication channel between the backend server and the RFID reader in the mobile RFID system is not assumed to be safe. Therefore it has become necessary to devise a new communication protocol that secures the privacy of mobile RFID-enabled devices. Recently, Lo *et al.* proposed a mutual key agreement protocol that secures the authenticity and privacy of engaged mobile RFID readers by constructing a secure session key between the reader and server. However, this paper shows that this protocol does not meet all of the necessary security requirements. Therefore we developed an enhanced mutual key agreement protocol for mobile RFID-enabled devices that alleviates these concerns. We further show that our protocol can enhance data security and provide privacy protection for the reader in an unsecured mobile RFID environment, even in the presence of an active adversary.

Keywords: Mobile RFID, Elliptic curve, Mutual authentication, Impersonation attack

1. Introduction

Radio Frequency Identification (RFID) is an automatic identification technology that allows remote interrogation of the ID data on RFID tags using radio frequency as a means of wireless communication between tagged objects and RFID readers. RFID technology has many applications, including more efficient material handling processes, the elimination of manual inventory counts, and the automatic detection of empty shelves and expired products in retail stores. RFID technology has a number of advantages over other identification technologies. It does not require a line-of-sight alignment, multiple tags can be almost simultaneously identified, and the tags do not destroy the integrity or aesthetics of the tagged object. The location of tagged objects can therefore be monitored automatically and continuously [1]. Recently, RFID technology has converged with mobile phone technology at a rapid pace; new applications are emerging that have begun to show the potential benefits of this convergence. Accompanying this convergence, however, are many security and privacy threats for mobile RFID-enabled devices. These threats include impersonation attacks, replay attacks, and the ability to trace the communicating parties. In view of such threats, the wide-spread adoption of mobile RFID requires proper protection [2]. Currently, RFID technologies are mainly designed for environments in which the RFID tags are mobile and the RFID readers are stationary. Therefore, most research into security and privacy issues has leaned toward the RFID tag owners rather than interrogating users carrying an RFID reader [3-8]. In mobile RFID, however, the RFID tags are stationary, the readers are mobile, and the communication environment is wireless and unsecure. Under this type of infrastructure, handheld devices, such as a mobile phone embedded with RFID reader modules, will be situated anywhere and operated using many RFID tags in various RFID application systems. It is difficult, without a novel communication protocol, to secure the privacy of a mobile RFID-enabled device. Therefore, there is need for new schemes to protect against privacy disclosure threats to mobile RFID readers.

In this paper we present an enhanced mutual key agreement protocol for mobile RFID-enabled devices. The proposed protocol is based on an elliptic curve cryptosystem [9-10] and can provide reader anonymity, forward security, impersonation resistance, and replay attack resistance in unsecure communication environments. In our scheme, a reader is embedded into a mobile phone and plays the role as a reader with a high computation ability. Even if an adversary eavesdrops on the communication between any two parties in the protocol, he/she does not obtain any information regarding the involved parties when using this protocol.

* Corresponding Author:

[1] Department of IT Convergence, Kyungil University /Gyeongsan, South Korea kambombomtonga@yahoo.com

[2] Department of Cyber Security, Kyungil University /Gyeongsan, South Korea ejyoon@kiu.ac.kr

Received March 21, 2012; Accepted June 15, 2012

The rest of the paper is organized as follows: in Section 2, we describe some of the preliminary work and notations that are used throughout this paper. We briefly review and cryptanalyze Lo *et al.*'s protocol and identify its inefficiencies in Section 3. In Section 4 we discuss how to overcome the inefficiencies of Lo *et al.*'s protocol by introducing the enhanced mutual key agreement protocol. In Section 5 we present the analyses that prove our protocol is secure against many types of attacks and also show that our protocol has a better performance than Lo *et al.*'s protocol by providing a comparison between the two. Finally, we draw our conclusions in Section 6.

2. Related Works

Many researchers have suggested many techniques to solve the problems found in mobile RFID systems [11-14]. Many of these studies, however, focused on enhancing the privacy and security of the mobile RFID tags and not the readers. But security and privacy enhancement on the reader side in mobile RFID system is more urgent, because the reader is not fixed, allowing anyone possessing the mobile device to obtain information from the tagged objects. In [15], Yang *et al.* proposed a lightweight dynamic identity based RFID authentication scheme to protect both the data privacy and the location privacy. However, Yang *et al.*'s scheme neglected the privacy disclosure threat to mobile RFID readers. Kim *et al.* [16] proposed the MARP (Mobile Agent for RFID Privacy Protection) protocol for high-level privacy protection. This protocol requires a public key center which manages the reader keys, a tag, a server, and a proxy. Lo *et al.* [17] proposed a mutual key agreement protocol based on an elliptic curve cryptosystem in order to enhance the content security of transmitted messages and provide entity anonymity to the mobile RFID reader. To ensure authenticity and privacy of engaged RFID readers during session key construction between the reader and server for unsecure wireless channels, Lo *et al.* introduced a trusted third party as the key distribution authority.

Recently, research in [18] showed that Lo *et al.*'s protocol is weak against insider impersonation attack. As a result, in this paper, we present an enhanced mutual key agreement protocol for mobile RFID enabled devices that overcomes the security and privacy weaknesses of Lo *et al.*'s protocol.

2.1 The Basic ECC Operations

First we must briefly introduce the basic mathematical operations regarding elliptic curves and the elliptic curve Diffie-Hellman key exchange algorithm. The ECC mathematical operations are defined over the elliptic curve $y^2=x^3+ax+b$, where $4a^3+27b^2\neq0$. All points (x, y) satisfying this equation plus a point at infinity lies on the elliptic curve. The security of the ECC depends on the difficulty of the elliptic curve Discrete Logarithm Problem. For example, let E be an elliptic curve defined over a finite field Fq and let $M \in E(Fq)$ be a point of order n. Given $N \in E(Fq)$, the elliptic curve discrete logarithm problem is to find the integer l, $0 \leq l \leq n-1$, so that $N = l.M$. Given M and N, it is computationally infeasible to obtain l if l is overly large. Here, l is called the discrete logarithm of N to the base M, making the main operation involved in the elliptic curve point multiplication achieved by:

a) Point addition: for $A_1= (x_{A1}, y_{A1})$ and $A_2= (x_{A2}, y_{A2})$ we can find a third point $A_3= (x_{A3,} y_{A3})$ on the curve such that $A_1+A_2=A_3$.

b) Point doubling: given a point $A_1= (x_{A1}, y_{A1})$ with $y_{A1}\neq0$, there exist a point $A_2= (x_{A2}, y_{A2})$ on the curve such that $A_2=2A_1$.

In our protocol, the Diffie-Hellman elliptic curve protocol (see Fig. 1) is used for session key sharing between the reader and server.

User A_1	User A_2
- Choose $d_v\in[1, n\text{-}1]$ - Compute: $S_v= d_v\ G$ - Send (S_v) - Receive (S_s) - Compute: $K= d_v\ S_{s=}\ d_v\ d_s\ G$	- Choose $d_s\in[1, n\text{-}1]$ - Compute: $S_s= d_s\ G$ - Receive: (S_v) - Send (S_s) - Compute: $K= d_s\ S_{v=}\ d_s\ d_v\ G$

Fig. 1. The Diffie-Hellman elliptic curve protocol

2.2 Notations

We assume a field F_q with size $q=p$, where p is large odd prime or $q=2^m$ in the case where q is a prime power. $G=(k_{g,}\ l_g)$ is the generator of the elliptic curve equation $Eq(a, b)$ which is over F_q:$k^2=k^3+al+b$(mod q), where $q>3$, a, $b\in\mathrm{F_q}$ and $4a^3+27b^2\neq0$(mod q). Two finite points $A_1=(k_1, l_1)$ and $A_2=(k_2, l_2)$ with order n, a large prime number, can be found in $E(\mathrm{F_q})$, where $A_1\neq\mathrm{O}$ and $A_2\neq\mathrm{O}$ (O denotes infinity). $Eq(a, b)$, $A_{1,}\ A_2$ and G are well known by all participating parties, which are the readers, backend server, and the trusted third party server in this case. We use the following notations throughout this paper:

- *Server*: backend application *Server*
- *TTP*: trusted third party
- *H*( ): one-way hash function
- $E_k(M)$: denotes the encryption of message M using key k
- *Cert* (¢): denotes a certificate
- ID_r: the identification of *Reader*
- ID_s: the identification of *Server*
- k_{auth}: the generated shared session key
- k_{sr}: the private key of *Reader*
- k_{pr}: the public key of *Reader*
- k_s: the public key of the *Server*
- k_{st}: the shared symmetric key between *Server* and *TTP*

- r: a nonce, the corresponding serial number of a generated key pair Y and X
- t_{r1},t_{r2},t_{r3}: the timestamp instances generated by *Reader*
- t_{r2_last}: the timestamp of the last incoming legitimate message sent from *Reader* and stored in *Server*
- k_t: the public key of *TTP*, k_t=i_1A_1+i_2A_2, where i_1 and i_2 are randomly selected from the interval [1, n-1]
- k_{ti}: the private key of *TTP*, k_{ti}=(i_1, i_2), where i_1 and i_2 are randomly selected from the interval [1, n-1] (Note: k_t and k_{ti} share the same values of i_1 and i_2)
- n_1, n_2: the random numbers generated at each session
- Y and X: the generated key pair used to protect the identification of a *Reader*, Y=x_1A_1+x_2A_2 and X=(x_1, x_2), where (x_1, x_2) are randomly selected from the interval [1, n-1]
- P_i: a randomly generated value (p_1, p_2) as the temporary pseudonym of an RFID *Reader* in a pre-defined period of time
- t_{r1_last}: the time stamp of the last incoming legitimate message sent from *Reader* and stored in *TTP*.
- ||: the concatenation operation.

3. Cryptanalysis of Lo *et al.*'s Protocol

Recently, Lo *et al.* proposed a mutual key agreement protocol for mobile RFID enabled devices. The goal of their protocol is that a mobile RFID reader acquires an identity-proved certificate from a trusted third party server in such a manner that the reader can communicate with the backend application server later with its pseudonym and the provable certificate to dynamically generate the session key for further message communication.

3.1 Review of Lo *et al.*'s Protocol

Lo *et al.* claimed that their protocol, shown in Fig. 2, is secure against common attacks even in presence of an active adversary in unsecured wireless communication environments. Normally in their protocol, a mobile RFID reader first requests a provable certificate *Cert*(X) with the legitimate key X from the *TTP*. When the *TTP* receives the request from the *Reader*, it generate a pair of keys (Y, X) via the elliptic curve cryptosystem and distributes them along with their corresponding certificates to the backend *Server* and the requesting *Reader*, respectively. Once the *Reader* obtains and verifies the secret key X through its certificate *Cert*(X), the *Reader* generates a pseudonym P_i to represent itself for the consequent communication to the backend *Server* instead of using its identification ID_r. A corresponding certificate *Cert*(P_i) is also created and transmitted to allow the backend *Server* to validate the pseudonym P_i. By exchanging the two sequentially generated values n_1G and n_2G, both the backend *Server* and the *Reader* construct the same authentication session key k_{auth}=n_1n_2G.

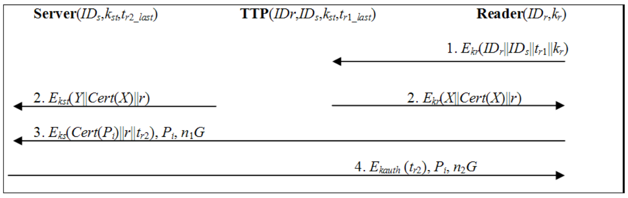
Server(ID_s,k_{st},t_{r2_last})
TTP(IDr,ID_s,k_{st},t_{r1_last})
Reader(ID_r,k_r)
1. $E_{kr}(ID_r||ID_s||t_{r1}||k_r)$
2. $E_{kst}(Y||Cert(X)||r)$
2. $E_{kr}(X||Cert(X)||r)$
3. $E_{ks}(Cert(P_i)||r||t_{r2})$, P_i, n_1G
4. E_{kauth} (t_{r2}), P_i, n_2G

Fig. 2. Lo *et al.*'s mutual key agreement protocol

3.2 An Impersonation Attack on Lo *et al.*'s Protocol

However, Mtonga and Yoon [18] showed that the protocol is weak against insider *Reader* impersonation attacks. An insider attacker who can access ID_r and ID_s from the data base can succeed in impersonating a legal *Reader* provided he generates a valid timestamp $t_{r1'}$> t_{r1_last} and secret key $k_{r'}$. Since $t_{r1'}$> $t_{r1\_last'}$, ciphertext $E_{kt}(ID_r||ID_s||t_{r1'}||k_{r'})$ is a valid message. This means that the *TTP* will respond accordingly. An attacker can successfully access the contents of the *TTP*'s response E_{kr}(X'||Cert(X')||r') by decrypting the message using its secrete key $k_{r'}$. Once an attacker accesses r', he can send E_{ks}(Cert($P_{i'}$)||r'||$t_{r2'}$),$P_{i'}$, $n_1'G$ to the *Server.* In this manner the attacker can easily wind up with the session key $k_{auth'}$=n_1'n_2'G. The whole attack is summarized in Fig. 3.

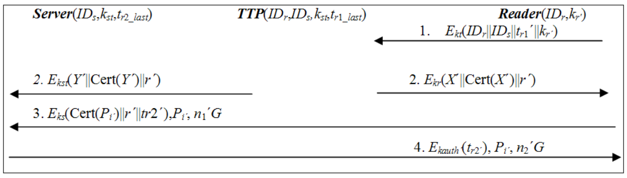
Server(ID_s,k_{st},t_{r2_last})
TTP(ID_r,ID_s,k_{st},t_{r1_last})
Reader(ID_r,$k_{r'}$)
1. $E_{kt}(ID_r||ID_s||t_{r1}'||k_{r'})$
2. E_{kst}(Y'||Cert(Y')||r')
2. E_{kr}(X'||Cert(X')||r')
3. E_{ks}(Cert($P_{i'}$)||r'||$tr2'$),$P_{i'}$, $n_1'G$
4. E_{kauth} ($t_{r2'}$), $P_{i'}$, $n_2'G$

Fig. 3. An impersonation attack on Lo *et al.*'s mutual key agreement protocol

4. The Enhanced Mutual Key Agreement Protocol

This section introduces an enhanced mutual key agreement protocol used to improve the security of Lo *et al.*'s protocol. The proposed protocol is based on an elliptic curve cryptosystem and can provide reader anonymity, forward security, impersonation resistance, and replay attack resistance in an unsecured communication environment. In our protocol, a reader is embedded into a mobile phone. We assume that the reader has the computation power to perform the elliptic curve cryptography algorithm and a large storage capacity, unlike the readers in previous RFID systems. Our protocol is a modification of Lo *et al.*'s protocol; it inherits some of the security and computational properties of Lo *et al.*'s protocol. Unlike [17], we introduce a public key cryptosystem between the *Reader* and the *TTP*. In our scheme, even if an adversary eavesdrops on the communication between the reader and the *TTP*, the adversary does not obtain any information regarding the involved parties. The protocol is summarized in Fig. 4.

(Note: in Fig. 4 k_{sr} is the private key of the *Reader* and k_{pr} is the public key).

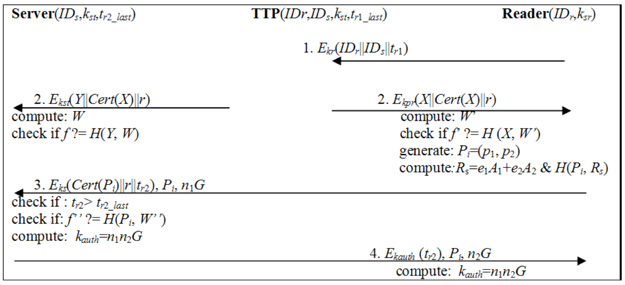
Server(ID_s,k_{st},t_{r2_last})
TTP(IDr,ID_s,k_{st},t_{r1_last})
Reader(ID_r,k_{sr})
1. $E_{kr}(ID_r\|ID_s\|t_{r1})$
2. $E_{kst}(Y\|Cert(X)\|r)$
compute: W
check if f ?= $H(Y, W)$
2. $E_{kpr}(X\|Cert(X)\|r)$
compute: W'
check if f' ?= $H(X, W')$
generate: P_i=(p_1, p_2)
compute: $R_s=e_1A_1+e_2A_2$ & $H(P_i, R_s)$
3. $E_{ks}(Cert(P_i)\|r\|t_{r2})$, P_i, n_1G
check if : $t_{r2}> t_{r2\_last}$
check if: f'' ?= $H(P_i, W'')$
compute: $k_{auth}=n_1n_2G$
4. E_{kauth} (t_{r2}), P_i, n_2G
compute: $k_{auth}=n_1n_2G$

Fig. 4. The enhanced mutual key agreement protocol

Step 1: Reader to TTP

By applying the asymmetric elliptic curve encryption operation [19] and using the *TTP* public key, the *Reader* protects its identity ID_r, identity of intended *Sever* ID_s, and valid (current) timestamp t_{r1}, and sends the message $E_{kt}(ID_r\|ID_s\|t_{r1})$ to the *TTP*.

Step 2: TTP to Reader Server

Upon receiving the cipher from the *Reader*, the *TTP* evaluates the validity of ID_r, ID_s, and t_{r1}. If $t_{r1}>t_{r1\_last}$, it implies that the message is fresh, whereas if $t_{r1}<t_{r1\_last}$, it implies that the message was already received in a previous session, thereby overcoming a replay attack. Once the received message is validated, via one-way hash function and elliptic curve computations, the *TTP* does the following:

- Generates a secrete key pair X=(x_1, x_2) and Y= $x_1A_1+x_2A_2$ and random value $R_t=i_1A_1+i_2A_2$, with (x_1, x_2) and (i_1, i_2) randomly selected from [1, n-1].
- Computes f=H(Y, R_t), f'=H(X, R_t)
- Construct certificates Cert(Y)=(f, V_{f1},V_{f2}) and Cert(X)= (f', V_{f2}, $V_{f2'}$). Where:

$$V_{f1} = i_1+f(\text{mod } n)$$
$$V_{f1'} = i_1+f'(\text{mod } n)$$
$$V_{f2} = i_2+f(\text{mod } n)$$
$$V_{f2'} = i_2+f'(\text{mod } n)$$

The *TTP* then sends cipher texts E_{kst}(Y‖Cert(Y) ‖r) and E_{kpr}(X‖Cert(X) ‖r) to the *Server* and *Reader*, respectively, where r is only the serial number of the generated key pair (X, Y). Finally, through a one way hash function, the *TTP* updates the stored time stamp as $t_{r1\_last}=t_{r1}$.

When the *Reader* and *Server* receive their corresponding messages from the *TTP*, they decrypt the message and validate secret keys X and Y using the corresponding certificates, Cert(X) for the *Reader* and Cert(Y) for the *Server*.

To validate secret key X, the *Reader* makes the following computations:

- Computes transient value W' using values from Cert(X) and known elliptic curve points A_1 and A_2 as:

$$\begin{aligned} W' &= V_{f1'}A_1+V_{f2'}A_2-f'(A_1+A_2) \\ &= (i_1+f'(\text{mod } n))A_1+(i_2+f'(\text{mod } n))f'(A_1+A_2) \\ &= A_1i_1+f'A1+i_2A_2+f'A_2-f'A_1-f'A_2 \\ &= A_1i_1+i_2A_2 \end{aligned}$$

- Checks whether the equation f' = H(X, W') holds. If the condition holds, then it means X is valid.

Similarly, *Server* also validates Y by:

- Calculating transient value W as:

$$\begin{aligned} W &= V_{f1}A_1+V_{f2}A_2-f(A_1+A_2) \\ &= (i_1+f(\text{mod } n))A_1+(i_2+f(\text{mod } n))A_2-fA_1\text{-}fA_2 \\ &= i_1A_1+fA_1+i_2A_1+fA_2\text{-}fA_1\text{-}fA_2 \\ &= i_1A_1+i_2A_2 \end{aligned}$$

- Checking if the condition f = H(Y, W) holds. If the condition holds, it means Y is valid.

Step 3: Reader to Server

After validating X, *Reader* then carries out the following computations:

- Generates pseudonym P_i=(p_1, p_2) which it uses to identify itself as a message destination in the future communication with the *Server*.
- Computes transient value $R_s=e_1A_1+e_2A_2$ where (e_1,e_2) is randomly selected from [1, n-1].
- Constructs a certificate Cert(P_i)=(f'', $V_{f1''}$, $V_{f2''}$) so that:

$$f''=H(P_i, R_s)$$
$$V_{f1''}=e_1+x_1f''(\text{mod } n)$$
$$V_{f2''}=e_2+x_2f''(\text{mod } n)$$

- Generates value n_1G as a contribution for the session key k_{auth} construction where n_1 is randomly selected from [1, n-1].

The *Reader* then utilizes the *Server*'s public key to encipher message E_{ks}(Cert(P_i) ‖r‖t_{r2}), P_i, n_1G and sends it to the *Server*.

Step 4: Server to Reader

When the *Server* receives the incoming message, it decrypts the cipher text and retrieves the certificate Cert(P_i), serial number r and timestamp t_{r2}. The *Server* then:

- Checks if r corresponds to the received key Y from *TTP* and
- Verifies pseudonym P_i with Cert(P_i) by checking if the condition f'' =H(P_i, W'') holds:

$$\begin{aligned} W'' &= V_{f1''}A_1+V_{f2''}A_2-f''Y \\ &= (e_1+x_1f''(\text{mod } n))A_1+(e_2+x_2f''(\text{mod } n))A_2 \\ &\quad -f''x_1A_1-f''x_2A_2 \\ &= e_1A_1+x_1f''A_1+e_2A_2+x_2f''A_2-f''x_1A_1-f''x_2A_2 \\ &= e_1A1+e_2A_2 \end{aligned}$$

The *Server* uses timestamp t_{r2} to control against replay attacks. If $t_{r2}<t_{r2\_last}$ it means that the received message is valid, whereas if $t_{r2}<t_{r2\_last}$, it means that the message was received in a previous interaction, so the *Server* does not respond. If the received message is valid, *Server* generates n_2 from [1, n-1], computes n_2G and authentication key $k_{auth}=n_2n_1G$. With the session key computed, the *Server* uses k_{auth} to encrypt t_{r2} and sends the message $E_{kauth}(t_{r2})$, n_2G, P_i to the *Reader*. Finally, the *Server* updates $t_{r2\_last}=t_{r2}$.

When the *Reader* receives the message from the *Server*, it uses P_i to identify itself as the message destination and computes $k_{auth}=n_2n_1G$, which it uses to decrypt and check if t_{r2} is the same as the one it generated. The established session key k_{auth} is maintained and will be used to encrypt and decrypt the messages between the *Reader* and *Server* in the normal operations of the RFID based systems.

Step 5: Reader to tag, tag to Reader and Reader to Server

With the session key established, the *Reader* can now interact with a tag by sending a challenge to the tag. When the tag responds, the *Reader* retrieves the pseudonym P_i and corresponding certificate Cert (P_i) and utilizes current session key k_{auth} to encrypt Cert (P_i), serial number r, current timestamp t_{r3}, and the tag's response value. After enciphering the message, the *Reader* sends the cipher text $E_{kauth}(\mathrm{Cert}(P_i)\ ||r||t_{r3}||\text{tag response})$, P_i to the *Server*. When the *Server* receives this message, it retrieves the current session key k_{auth} based on the received pseudonym P_i and decrypts the cipher text to obtain the certificate Cert(P_i),serial number r, current time stamp t_{r3}, and the response from the tag. According to the retrieved serial number r, the *Server* can find the corresponding secret key Y to verify pseudonym P_i with Cert (P_i). The *Server* also checks timestamp t_{r3} to prevent a replay attack. If the verified message is valid, the *Server* trusts that the tag's response came from the authenticated *Reader* even though the *Reader* does not reveal its real identity. With the *Reader* accepted, the *Server* can now perform secure communication accordingly [19, 20].

5. Analysis

In this section, we describe several common attacks applicable to mobile RFID systems. We also show how the proposed protocol resists these attacks.

5.1 Security Analysis

a) *Impersonation Attack*

In this type of attack, the attacker impersonates a legitimate user and forges the authentication message using the information obtained from the sessions. In unsecured communication environments, an attacker can intercept the transmitted messages and try to impersonate any legal party in the protocol, however, without the knowledge of secrete key k_{st}, private key k_{sr} and k_{ti}, the attacker cannot obtain any information from the encrypted messages. In addition, for an insider attacker who can access identities ID_r and ID_s from the data base and wants to carry out an insider impersonation attack, if he/she can successfully generate valid timestamp t_{r1}'$>t_{r1\_last}$ and send $E_{kt}(ID_r||ID_s||t_{r1}')$ *to TTP*, he/she cannot decrypt the response $E_{kpr}(R_c||\mathrm{Cert}(R_c)\ ||r)$ from *TTP* due to the public key cryptosystem between the *Reader* and the *TTP*. Therefore, our protocol is secure against impersonation attacks.

b) *Replay Attack*

This is an attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. In the presented protocol above, the timestamp generated at each session can be used to detect this type of attack. Suppose an attacker succeeds to capture $E_{kt}(ID_r||ID_s||t_{r1})$ and try to resend at time t^*:

- If $t^*>t_{r1}=t_{r1\_last}$ then the *TTP* responds
- If $t^*<t_{r1}= t_{r1\_last}$, *TTP* aborts the session

In our protocol all of the parties in the scheme maintain the time stamps of the last successful processed legitimate message. Upon receiving a message from the *Reader*, the *TTP* and *Server* always first decrypt and check to see if the value of the received timestamp is less than or greater than the stored value; if the timestamp is less than the stored value, then a replay attack is detected. Also, an attacker has no access to secret key k_{st} and private keys k_{ti}, k_{sr}, and k_{auth}, therefore he/she cannot access the timestamps encrypted with these keys. This makes it almost impossible for an attacker to counterfeit a legitimate message and spoof any party in the scheme. As such, the protocol is secure against replay attacks.

c) *Forward Security*

The forward-security property means that even if the adversary obtains the current secret key, he still cannot derive the keys used for past time periods. In the protocol, the session key generation processes involves a randomly generated n_1 and n_2 and so are one-time valid for each session. Therefore, even if the *Reader* is compromised during session j, the attacker cannot derive previous authentication session keys without knowledge of the n_1 and n_2 generated in the previous session.

d) *Session Key Security*

If an attacker can intercept message 3 and 4 in the protocol, and get n_1G and n_2G, he/she cannot succeed in computing session key $k_{auth}=n_1n_2G$ without knowledge of random values n_1 and n_2 because of the hardness of the ECDLP [10].

e) *Reader Anonymity and Un-traceability*

The anonymity property implies that the real identity of *Reader* ID_r must be unknown. Un-traceability means that the equality or inequality of two *Readers* must be impossible to determine. In our protocol, no one except the *TTP* acquires the true identity of a *Reader*. At the same time, a distinct enciphered message is generated for each session before the *Reader* sends it as a request to the *TTP*. This ensures that ID_r remains anonymous to all parties except the TTP. To identify itself as message recipient, each *Reader* uses the pseudonym P_i instead of its true identity ID_r. This means, the *Server* only recognizes the pseudonym P_i. This ensures the anonymity of the *Reader*. To ensure un-traceability, the pseudonym P_i and Cert(P_i) can be made one-time valid.

5.2 Performance Analysis

The differences between Lo *et al.*'s mutual key agreement and our modified mutual key agreement protocols are summarized in terms of security, privacy and computational feasibility on the *Reader* side in Table 1.

Table 1. Performance Comparisons of Le *et al.*'s Protocol and Proposed Protocol

Protocols Attacks and Properties	Lo *et al.*'s MKA Protocol	Proposed MKA Protocol
Replay attack	Yes	Yes
Forward security	Yes	Yes
Anonymity & non-repudiation	Yes	Yes
Impersonation attack	No	Yes
Computation overload for reader	No	Yes
Session key security	Yes	Yes

6. Conclusion

Recently, RFID technology has been rapidly converging with mobile phone technology, and new applications are emerging that begin to show the potential benefit of this convergence. With this convergence, many security and privacy threats for mobile RFID-enabled devices have become evident. Therefore, when designing a secure mobile RFID protocol, one must always consider protection against these security threats. In this paper we present a protocol that provides for secure session key generation between mobile RFID-*Reader*-equipped phones and a *Server*. Our protocol is secure against the common security threats found in mobile RFID, such as impersonation attacks, replay attacks, the tracing of communicating parties, and preserves *Reader* privacy in unsecured wireless communication environments. We have also shown that our protocol is computationally feasible for the *Reader* since the *Reader* inner protocol only needs to concatenate three parameters before sending the request unlike Lo *et al.*'s protocol where the *Reader* concatenates four message entries.

References

[1] R. Kumar, "Framework of Smart Mobile RFID Networks", *Journal of Information Engineering and Applications*, Vol. 1, No.1, pp. 13-21. 2011. Article (CrossRef Link)

[2] H-M. Sun, S-M. Chen, C-E. Lu, and C-T. Yang "Secure and Efficient Mobile RFID Authentication Protocol", *Journal of Computers*, Vol. 20, No. 2, pp. 68-83, 2007. Article (CrossRef Link)

[3] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal Re-encryption for Mixnets," *in Proc. of CT-RSA 2004*, *LNCS*, Vol. 2964, pp. 163–178, 2004. Article (CrossRef Link)

[4] S. Weis, S-E. Sarma, R-L. Rivest, and D-W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency identification Systems," *in Proc. of SPC 2004*, *LNCS*, Vol. 2802, pp. 50–59, 2004. Article (CrossRef Link)

[5] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags," *in Proc. of RFID Privacy Workshop*, 2003. Article (CrossRef Link)

[6] J. Saito, J-C. Ryou, and K. Sakurai, "Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags", *in Proc. of EUC 2004*, *LNCS*, Vol. 3207, pp. 55-84, 2004. Article (CrossRef Link)

[7] H. Dirk, and M. Paul, "Hash-based Enhancement of Location Privacy for Radio-frequency Identification Devices using Varying Identifiers", *in Proc. of IEEE PerCom 2004*, pp. 149-153, 2004. Article (CrossRef Link)

[8] H. Chien, and C. Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards", *Computer Standards and Interfaces*, Vol. 29, No. 2, pp.254–259, 2007. Article (CrossRef Link)

[9] F. Yu, H. Kuo, L. Feipei, and C. Shyong, "ID-based Digital Signature Scheme on the Elliptic Curve Cryptosystem," *Computer Standards and Interfaces*, Vol. 29, No.6, pp.601–604, 2007. Article (CrossRef Link)

[10] A. Miller, "Use of elliptic curves in cryptography", *in Proc. of Advances in Cryptology 1986*, *LNCS*, Vol. 218, pp. 417–26, 1986. Article (CrossRef Link)

[11] A. Juels, P. Syverson, and D. Bailey, "High-Power Proxies for Enhancing RFID Privacy and Utility," *in Proc. of PET 2006*, *LNCS*, Vol. 3856, pp. 210–216, 2006. Article (CrossRef Link)

[12] J. Lee, D. Choi, and H. Kim, "Practical Privacy Protection Mechanism for Networked RFID Environment," *in Proc. of IWSEC 2006*, 2006. Article (CrossRef Link)

[13] M. Rieback, B. Crispo, and A. Tanenbaum, "RFID Guardian: A Battery- powered Mobile Device for RFID Privacy Management," *in Proc. of ACISP 2005*,

LNCS, Vol. 3574, pp. 184-194, 2005. Article (CrossRef Link)

[14] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems," *in Proc. of SPC 2004*, *LNCS*, Vol. 2802, pp. 201-212, 2004. Article (CrossRef Link)

[15] M. Yang, J. Wu, and S. Chen, "Protect Mobile RFID Location Privacy using Dynamic Identity," *in Proc. of IEEE ICCI 2008*, pp. 366–374, 2008. Article (CrossRef Link)

[16] S. Kim, S. Yeo, and S. Kim, "MARP: Mobile Agent for RFID Privacy Protection," *in Proc. of CARDIS 2006*, *LNCS*, Vol. 3928, pp. 300-312, 2006. Article (CrossRef Link)

[17] N. Lo, K. Yeh, and C. Yeun, "New Mutual Agreement Protocol to Secure Mobile RFID-enabled Devices," *Information Security Technical Report*, Vol. 13, No. 3, pp. 151–157, 2008. Article (CrossRef Link)

[18] K. Mtonga, and E.-J. Yoon, "Cryptanalysis of Lo et al.'s Mutual Key Agreement Protocol for Mobile RFID-Enabled Devices", *in Proc. of IEEK Summer Conference*, pp.165-168, 2012. Article (CrossRef Link)

[19] A. Menezes, and S. Vanstone, "Elliptic Curve in Cryptosystem and Their Implementation," *Journal of Cryptology*, Vol. 6, No. 4, pp. 209–224, 1993. Article (CrossRef Link)

[20] H. Chien, and C. Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards", *Computer Standards and Interfaces*, Vol. 29, No. 2, pp. 254–259, 2007. Article (CrossRef Link)

Kambombo Mtoga received his B.S. degree in Mathematics Education from University of Malawi-Chancellor College, Malawi, in 2010. Currently, he is a graduate student of Department of IT Convergence at the Kyungil University, Republic of Korea. His current research interests are cryptography, authentication technologies, and RFID security.

Eun-Jun Yoon received his MSc degree in computer engineering from Kyungil University in 2002 and the PhD degree in computer science from Kyungpook National University in 2006, Republic of Korea. From 2007 to 2008, he was a full-time lecturer at Faculty of Computer Information, Daegu Polytechnic College, Republic of Korea. From 2009 to 2011, he was a 2nd BK21 contract professor at the School of Electrical Engineering and Computer Science, Kyungpook National University, Republic of Korea. Currently, he is a professor at the Department of Cyber Security, Kyungil University, Republic of Korea. His current research interests are cryptography, authentication technologies, smart card security, network security, mobile communications security, and steganography.