A Privacy and Delegation Enhanced User Authentication Protocol for Portable Communication Systems

Ren-Chiun Wang*, Wen-Shenq Juang†, and Chin-Laung Lei*‡
*Department of Electrical Engineering
National Taiwan University
Email:rcwang@fractal.ee.ntu.edu.tw, lei@cc.ee.ntu.edu.tw
†Department of Information Management
National Kaohsiung First University of Science and Technology
Email: wsjuang@ccms.nkfust.edu.tw
‡Corresponding author.

***Abstract*—In 2005, a delegation-based authentication protocol for portable communication systems was proposed by Lee and Yeh. The major merits include: 1. the identity of mobile user is not exposed over an open network; 2. the mobile user can construct the digital signature for roaming service requests by himself; 3. the protocol satisfies secrecy, authenticity, data integrity, and non-repudiation properties; 4. the mutual authentication between the mobile user (MS) and the visited location register (VLR) is satisfied; and 5. the computation and communication cost is low. Later, Lee *et al.* showed that a valid malicious VLR can trick the home location register (HLR) by forging authentication messages and overcharging the service fee in Lee-Yeh's protocol. At the same time, Lee *et al.* proposed an improved method to enhance the security and the efficiency. The intentions of this paper include: 1. to demonstrate that both Lee-Yeh and Lee *et al.*'s protocols do not keep the privacy of MS actually; 2. to show that the overcharge problem still exists in Lee *et al.*'s protocol; and 3. to propose a new method which can enhance the delegation and security level and keep the privacy and the efficiency of MS.**

***Index Terms*—Authentication; Hash Function; Privacy; Proxy signature; Wireless Communications.**

I. Introduction

The portable communication system (PCS) is a convenient way for subscribers to obtain desired services from service providers without using any physical circuits. Oppositely, the radio waves transmitted way is not secure since anyone can easily eavesdrop the contents of communications from air. A widely adopted way is to employ cryptosystems to provide secrecy, authenticity, data integrity, and non-repudiation features.

Many well-known public key cryptosystems can be adopted to provide the above features [1], [2], [3]. However, the speed of encryption and decryption in public key cryptosystems is lower than secret key cryptosystems such as AES [4]. Also, the public key need to be changed periodically. The scalability, the communication bandwidth, the computation capability and the storage space are inherent fatal in resource-constrained wireless environments and portable devices. Therefore, the cost-benefit analysis and the performance are major concerns to participants in PCSs.

Global System for Mobile Communication (GSM) is a popular standard for the mobile stations in the world [5]. Based on the concept of the challenge/response technique in secret key cryptosystem, the computation cost of MS do not increase dramatically and the long-term secret key K_i is embedded in the SIM card. We briefly demonstrate the GSM protocol in Figure 1, where IMSI is the international mobile subscriber identity, TMSI is the temporary mobile subscriber identity, LAI is the location area identity, and RAND is the random number. In the protocol, the non-repudiation property is not provided, so a dishonest user may deny the calls. Besides, the privacy of user identity is not protected due to the real identity IMSI is exposed over open networks and there are no protection

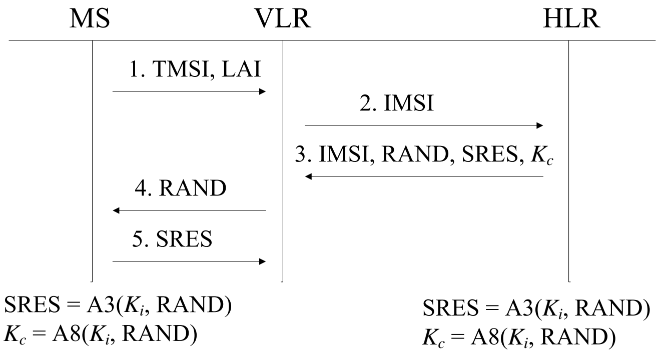


Fig. 1. The authentication in GSM systems

mechanisms between VLR and HLR. The sensitive information could be eavesdropped. Finally, the mutual authentication property between MS and VLR is also not provided.

In 2005, Lee and Yeh employed the concept of proxy signature to propose a delegation-based authentication protocol [6]. In which, many admired requirements are achieved such as identity privacy, non-repudiation, mutual authentication between MS and VLR, easy key management, low computation cost, and the communication efficiency. Unfortunately, Lee *et al.* pointed out that Lee and Yeh's protocol is not secure against a valid malicious VLR from forging service request witnesses without the help of MS [7]. The overcharge problem happens. At the same time, Lee *et al.* employed the concept of hashing chain [8] to propose an improved method for eliminating the above weakness. In Lee *et al.*'s protocol, it not only keeps the same requirements, but also enhances the computation efficiency by the pre-computation technique.

The major contributions of this paper include: (1) to demonstrate that Lee-Yeh and Lee *et al.*'s protocols do not keep the privacy of user identity actually; (2) to show that the overcharge problem still exists in Lee *et al.*'s protocol; (3) to propose a novel method to keep Lee *et al.*'s requirements and to enhance the efficiency and the privacy of MS.

In next section, we review Lee *et al.*'s improvement and show their weakness. In Section 3, we present our method. In Section 4, we analyze the security of the proposed protocol. In Section 5, we analyze the efficiency of our proposed protocol and the related protocols. Finally, we conclude this paper in Section 6.

II. Lee *et al.*'s Delegation-Based Authentication Protocol

In this section, we briefly review Lee *et al.*'s protocol [7], demonstrate the linkability of MS's identity and show that the overcharge problem exists in their protocol.

A. Protocol

The protocol consists of on-line and off-line authentication processes. In the on-line authentication process, VLR verifies the signature of the service request and connects HLR for obtaining the first session key on demand. In the off-line authentication process, VLR does not need to connect HLR frequently for asking next verifier when MS accesses the network via VLR again. Based on the concept of hashing chain [8], VLR can identify MS by using the old token to generate the next token simultaneously.

1) **Parameters.** p is a large prime number (512 bits); q is a prime factor of $p-1$ (160 bits); g is a generator in group Z_p^*; K_{VH} is a pre-shared long-term secret key between VLR and HLR; ID_V and ID_H denote the identities of VLR and HLR; $[m]_K$ denotes the message m encrypted using the key K in a symmetric encryption scheme; $m_1 \| m_2$ denotes the concatenation of two random strings; and h() is a one-way hash function in cryptography.
2) **Setup.** HLR generates a private/public key pair (x, v), where x is a random number and v = g^x mod p. For each registered MS, HLR selects a random number k and calculates K = g^k mod p and σ = $x + kK$ mod q, where (σ, K) is the key pair shared between MS and HLR and K is the pseudonym of MS. After that, HLR writes (σ, K) into MS's SIM card and stores them with the real identity of MS into a secure database.
3) **Pre-compute.** MS generates a random number n_1, calculates a hashing chain $\text{h}^1(n_1)$, $\text{h}^2(n_1)$, ..., $\text{h}^{n+1}(n_1)$ and stores them, where n is a pre-defined constant used for limiting the times of the off-line authentication.
4) **On-line authentication.**
   1. MS sends K to VLR.

2. VLR generates a random number n_2 and sends it with ID_V back.
3. (a) MS generates a signature (r, s) for the message N_1, n_2 and ID_V, where t is a random number, r = g^t mod p and s = ($\sigma *$ h($N_1 \parallel n_2 \parallel ID_V$) + $t * r$) mod q.

   (b) MS sends (r, s, K, N_1, ID_H, ID_V) to VLR.
4. (a) VLR verifies whether the equation $g^s = (vK^K)^{h(N_1 \| n_2 \| ID_V)} r^r$ mod p holds. If the verification is successful, VLR encrypts the message (N_1, n_2, ID_V) by using the key K_{VH}.

   (b) VLR sends $[N_1 \parallel n_2 \parallel ID_V]_{K_{VH}}$ with ID_H and ID_V to HLR.
5. (a) HLR decrypts the message $[N_1 \parallel n_2 \parallel ID_V]_{K_{VH}}$ by using the key K_{VH}. According to K, HLR searches the corresponding key σ in its database. Then HLR calculates the first session key C_1 = h($N_1 \parallel n_2 \parallel n_3 \parallel \sigma$), where n_3 is also a random number.

   (b) HLR sends the encrypted message $[[N_1, n_3, ID_V]_\sigma \parallel n_2 \parallel l \parallel C_1]_{K_{VH}}$ with ID_H and ID_V to VLR, where l = N_1.
6. (a) VLR decrypts $[[N_1, n_3, ID_V]_\sigma \parallel n_2 \parallel l \parallel C_1]_{K_{VH}}$ to obtain $[N_1, n_3, ID_V]_\sigma$, n_2, l and C_1.

   (b) If n_2 and l are the same as before, VLR sets up the first time session key SK = C_1.

   (c) VLR forwards $[N_1, n_3, ID_V]_\sigma$ with ID_V to MS.

   (d) MS decrypts $[N_1, n_3, ID_V]_\sigma$ to obtain N_1, n_3 and ID_V. If N_1 is correct, MS believes that VLR is authenticated by HLR.

   (e) MS calculates C_1 = h($N_1 \parallel n_2 \parallel n_3 \parallel \sigma$) and sets up it as the current session key SK.

5) ***i*-th Off-line authentication.**

a) MS picks the verifier h$^{n-i+1}(n_1)$ from the database and encrypts it by using the key C_i. MS sends the encrypted result to VLR.

b) VLR decrypts [h$^{n-i+1}(n_1)$]$_{C_i}$, checks the counter $i < n$, and verifies whether the digest of the decrypted message is equal to l. If they are correct, MS is authenticated. VLR renews the verifier l = h$^{n-i+1}(n_1)$ for next authentication and the counter i = i + 1 and calculates next session key SK = C_{i+1} = h(l, C_i). We demonstrate the protocol in Figure 2.

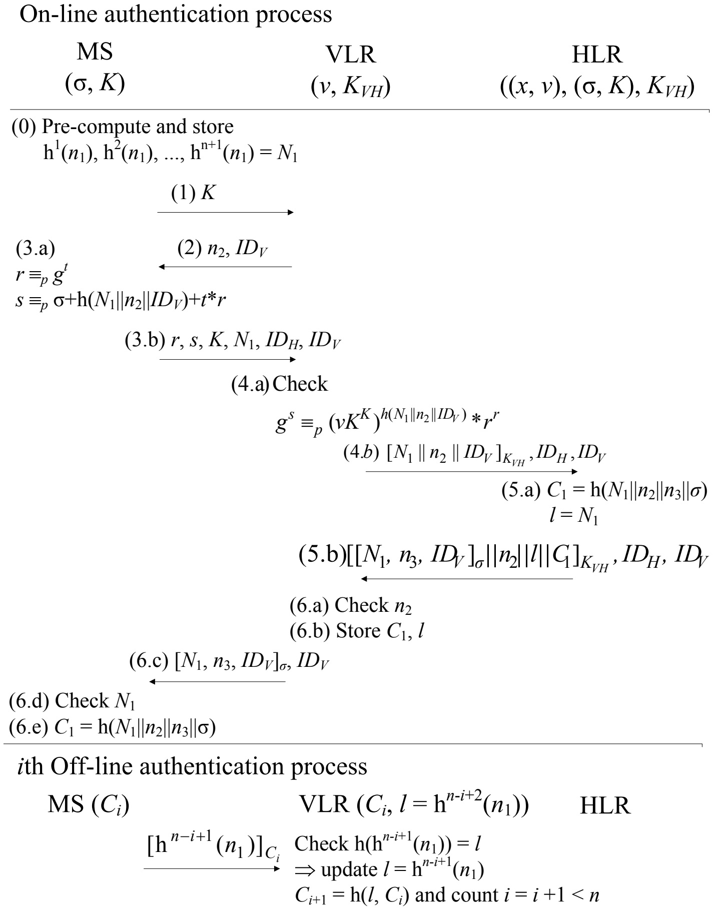

On-line authentication process
MS
(σ, K)
VLR
(v, K_VH)
HLR
((x, v), (σ, K), K_VH)
(0) Pre-compute and store
h^1(n1), h^2(n1), ..., h^n+1(n1) = N1
(1) K
(3.a)
r ≡p g^t
s ≡p σ+h(N1||n2||IDV)+t*r
(2) n2, IDV
(3.b) r, s, K, N1, IDH, IDV
(4.a) Check
g^s ≡p (vK^K)^h(N1||n2||IDV) * r^r
(4.b) [N1 || n2 || IDV]KVH, IDH, IDV
(5.a) C1 = h(N1||n2||n3||σ)
l = N1
(5.b)[[N1, n3, IDV]σ||n2||l||C1]KVH, IDH, IDV
(6.a) Check n2
(6.b) Store C1, l
(6.c) [N1, n3, IDV]σ, IDV
(6.d) Check N1
(6.e) C1 = h(N1||n2||n3||σ)
ith Off-line authentication process
MS (Ci)
VLR (Ci, l = h^n-i+2(n1))
HLR
[h^n-i+1(n1)]Ci
Check h(h^n-i+1(n1)) = l
⇒ update l = h^n-i+1(n1)
Ci+1 = h(l, Ci) and count i = i +1 < n

Fig. 2. Lee *et al.*'s scheme

B. Privacy of MS

Both of Lee-Yeh and Lee *et al.*'s protocols use the pseudonym K to replace the real identity IMSI and no one can derive the relationship between K and IMSI. However, the pseudonym K is never changed after MS sends the service request.

It means that VLR and other eavesdroppers can easily trace to the same MS when the service request was sent. Therefore, we say that the trajectory protection of MS is not enough in Lee-Yeh and Lee *et al.*'s protocols.

C. The Overcharge Problem

After MS has visited a valid malicious VLR, VLR can forge authentication messages by the help of other MS. It implies that VLR still can trick HLR to charge double or more service fees in Lee *et al.*'s protocol. We show a simple example and assume that MS_1 has visited VLR, MS_n is requiring the personal service and both of MS_1 and MS_n have registered to the same HLR. Note that MS_1 and MS_n do not need to register to the same HLR in this problem.

1) After MS_1 has visited VLR, VLR records the pseudonym K_1.
2) When MS_n sends the service request to run the on-line authentication process, VLR sends the messages $[N_1 \parallel n_2 \parallel ID_V]_{K_{VH}}$ and $[N_1 \parallel n_2' \parallel ID_V]_{K_{VH}}$ to HLR in parallel, where n_2 and n_2' are random numbers and n_2' is used to forge MS_1's service request. Note that N_1 is chosen by MS_n.
3) Without loss of generality, VLR will receive $[[N_1, n_3, ID_V]_{\sigma_n} \parallel n_2 \parallel l \parallel C_1]_{K_{VH}}$ and $[[N_1, n_3', ID_V]_{\sigma_1} \parallel n_2' \parallel l' \parallel C_1']_{K_{VH}}$ from HLR, where n_3 and n_3' are random numbers and (n_3', C_1') is response for MS_1 service request. σ_1 and σ_n are MS_1 and MS_n's secret keys, respectively.
4) VLR stores (C_1, l) and (C_1', l') and waits to run the off-line authentication process with MS_n. Note that l is equal to l'.
5) In the i-th off-line authentication process, MS_n encrypts the verifier $\text{h}^{n-i+1}(n_1)$ by using the key C_i and sends the encrypted result to VLR. Without loss of generality, VLR renews the verifier $l = \text{h}^{n-i+1}(n_1)$ for the next authentication of MS_n and the counter $i = i + 1$ and calculates the next session key $C_{i+1} = \text{h}(l, C_i)$.

   At the same time, VLR can forge the next verifier $l' = \text{h}^{n-i+1}(n_1)$ of MS_1 and the session key $C_{i+1}' = \text{h}(l', C_i')$. It means that VLR can forge m times the service request witnesses of MS_1 after MS_n has visited m times VLR. VLR imitates successfully MS_1's service request to trick HLR of MS_1 for charging the roaming fee without the knowledge of MS_1's secret key σ_1. The attack is also shown in Figure 3.

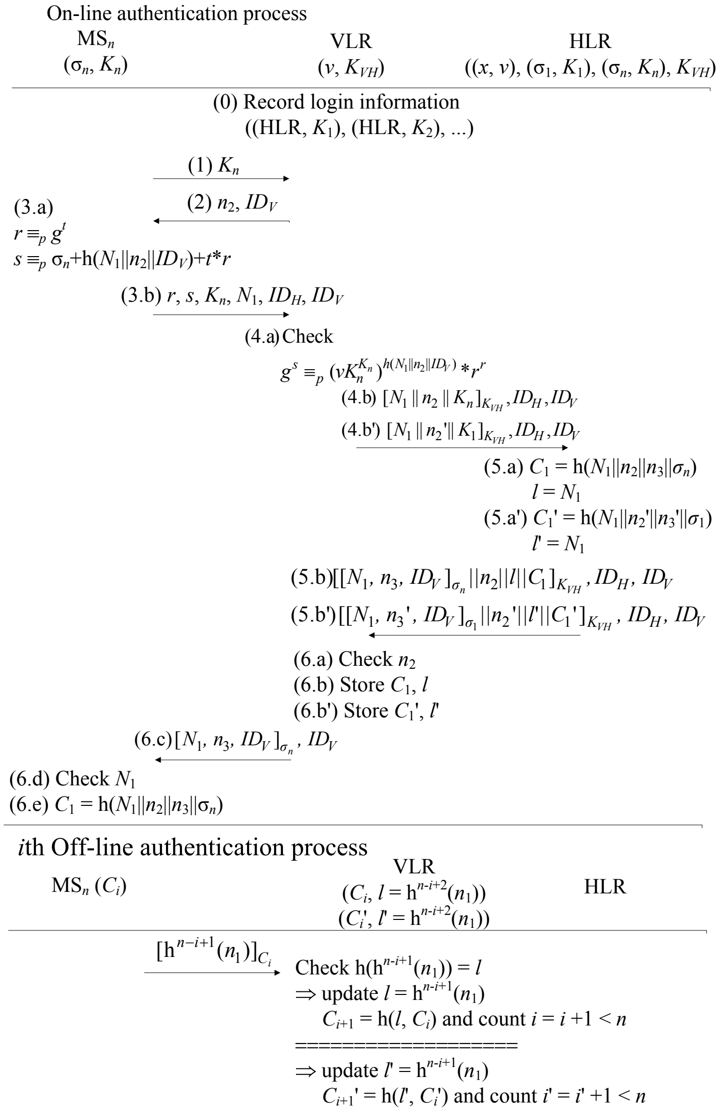

On-line authentication process
MS_n
(σ_n, K_n)
VLR
(v, K_{VH})
HLR
$((x, v), (\sigma_1, K_1), (\sigma_n, K_n), K_{VH})$
(0) Record login information
((HLR, K_1), (HLR, K_2), ...)
(1) K_n
(2) n_2, ID_V
(3.a)
$r \equiv_p g^t$
$s \equiv_p \sigma_n + \text{h}(N_1 \| n_2 \| ID_V) + t * r$
(3.b) $r, s, K_n, N_1, ID_H, ID_V$
(4.a) Check
$g^s \equiv_p (vK_n^{K_n})^{h(N_1 \| n_2 \| ID_V)} * r^r$
(4.b) $[N_1 \| n_2 \| K_n]_{K_{VH}}, ID_H, ID_V$
(4.b') $[N_1 \| n_2' \| K_1]_{K_{VH}}, ID_H, ID_V$
(5.a) $C_1 = \text{h}(N_1 \| n_2 \| n_3 \| \sigma_n)$
$l = N_1$
(5.a') $C_1' = \text{h}(N_1 \| n_2' \| n_3' \| \sigma_1)$
$l' = N_1$
(5.b) $[[N_1, n_3, ID_V]_{\sigma_n} \| n_2 \| l \| C_1]_{K_{VH}}, ID_H, ID_V$
(5.b') $[[N_1, n_3', ID_V]_{\sigma_1} \| n_2' \| l' \| C_1']_{K_{VH}}, ID_H, ID_V$
(6.a) Check n_2
(6.b) Store C_1, l
(6.b') Store C_1', l'
(6.c) $[N_1, n_3, ID_V]_{\sigma_n}, ID_V$
(6.d) Check N_1
(6.e) $C_1 = \text{h}(N_1 \| n_2 \| n_3 \| \sigma_n)$
ith Off-line authentication process
MS_n (C_i)
VLR
$(C_i, l = \text{h}^{n-i+2}(n_1))$
$(C_i', l' = \text{h}^{n-i+2}(n_1))$
HLR
$[\text{h}^{n-i+1}(n_1)]_{C_i}$
Check $\text{h}(\text{h}^{n-i+1}(n_1)) = l$
$\Rightarrow$ update $l = \text{h}^{n-i+1}(n_1)$
$C_{i+1} = \text{h}(l, C_i)$ and count $i = i + 1 < n$
====================
$\Rightarrow$ update $l' = \text{h}^{n-i+1}(n_1)$
$C_{i+1}' = \text{h}(l', C_i')$ and count $i' = i' + 1 < n$

Fig. 3. The overcharge problem in Lee *et al.*'s protocol

III. Our Protocol

In this section, we propose a new method to overcome the linkability and the overcharge problems. The used parameters are the same as Lee *et al.*'s protocol.

1) **Setup.** HLR generates a private/public key pair (x, v), where x is a random number and $v = g^x \bmod p$. For each registered MS, HLR selects a random number k and calculates $K = g^k \bmod p$ and $\sigma = (-k^{-1}xK) \bmod q$, where σ is the secret key shared between MS and HLR and K is the pseudonym of MS. After

that, HLR writes (σ, S_{Temp} = NULL, K) into MS's SIM card and stores (k, K, σ) with the real identity of MS into a secure database.

2) **Pre-compute.** MS generates a random number n_1 and calculates a hashing chain h^1(n_1), h^2(n_1), ..., h^{n+1}(n_1) = N_1 and K_{new} = $K^{h(N_1\|\sigma)}$ mod p. If s_{Temp} is null, MS calculates s_{Next} = ($\sigma * K^{-1} * h(N_1\|\sigma)^{-1} * K_{new}$) mod q; otherwise, MS calculates s_{Next} = ($s_{Temp} * K^{-1} * h(N_1\|\sigma)^{-1} * K_{new}$) mod q. MS then selects a random number t, computes r = g^t mod p, and stores all the computed results.

3) **On-line authentication.**

   1. MS sends K to VLR.
   2. VLR generates a random number n_2 and sends it with ID_V back.
   3. (a) MS generates a signature s for the messages h(N_1 $\|$ n_2 $\|$ ID_V), where if s_{Temp} = NULL, then s = (σ + h(N_1 $\|$ n_2 $\|$ ID_V) + $t * r$) mod q; otherwise, s = (s_{Temp} + h(N_1 $\|$ n_2 $\|$ ID_V) + $t * r$) mod q.

      (b) MS sends s with (n_2, r, N_1, ID_H, ID_V) to VLR.
   4. (a) VLR verifies whether the equation $g^{h(N_1\|n_2\|ID_V)} r^r$ = ($v^K K^s r^r$) mod p holds or not. If the verification is successful, VLR believes that MS is a privileged user.

      (b) VLR encrypts the message (N_1, n_2, K) by using the key K_{VH} and sends the encrypted result with ID_H and ID_V to HLR.
   5. (a) HLR decrypts the message $[N_1 \| n_2 \| K]_{K_{VH}}$ by using the key K_{VH}. According to K, HLR searches the corresponding key σ in its database. Then HLR calculates the session key C_1 = h(N_1 $\|$ n_2 $\|$ K_{new} $\|$ σ) and replaces K with K_{new}, where K_{new} = $K^{h(N_1\|\sigma)}$ mod p.

      (b) HLR sends the encrypted message [h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ) $\|$ n_2 $\|$ l $\|$ C_1]$_{K_{VH}}$ with ID_H and ID_V to VLR, where l = N_1.
   6. (a) VLR decrypts [h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ) $\|$ n_2 $\|$ l $\|$ C_1]$_{K_{VH}}$ to obtain h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ), n_2, l and C_1. If n_2 and l are the same as before, VLR sets up the first time session key SK = C_1.

      (b) VLR stores (C_1, l) for the off-line authentication.

      (c) VLR forwards h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ) with ID_V to MS.

      (d) MS verifies whether the received digest value is the same as h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ). If it is correct, MS believes that VLR is authenticated by HLR.

      (e) MS calculates C_1 = h(N_1 $\|$ n_2 $\|$ K_{new} $\|$ σ), sets up it as the current session key SK and replaces (K, S_{Temp}) with (K_{new}, S_{Next}).

4) i**-th Off-line authentication.** The process is the same as Lee *et al.*'s protocol.

IV. Discussion

A. Security Analysis

We analyze that the proposed protocol is secure against some well-known security threats.

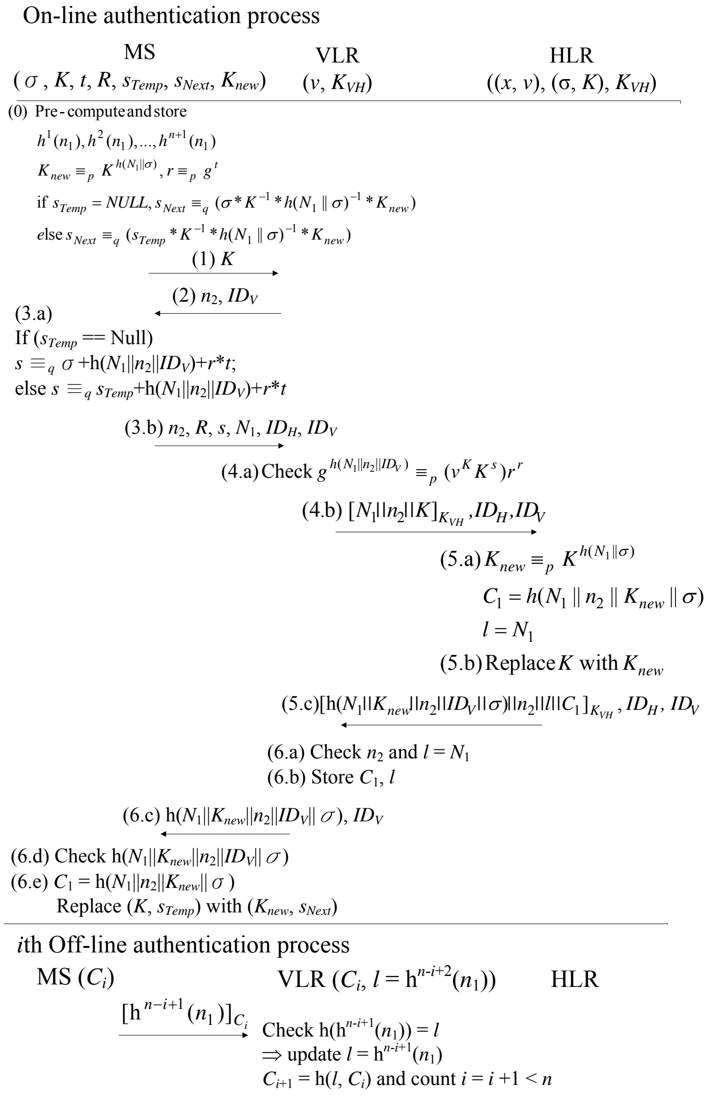
1) **Mutual authentication.** The goal of the mutual authentication is to establish an agreed session key SK between MS and VLR. In our protocol, the task will be finished by the help of HLR. Let MS $\overset{SK}{\longleftrightarrow}$ VLR denote that MS shares a secret key SK with VLR. The mutual authentication is complete between MS and VLR if there is a session key SK such that MS believes MS $\overset{SK}{\longleftrightarrow}$ VLR, and VLR believes MS $\overset{SK}{\longleftrightarrow}$ VLR. A strong mutual authentication may lead to the following statement:

   a) MS believes that VLR believes MS $\overset{SK}{\longleftrightarrow}$ VLR, and

   b) VLR believes that MS believes MS $\overset{SK}{\longleftrightarrow}$ VLR.

   By the help of HLR, MS and VLR can do mutual authentication in the on-line authentication process as follows.

   a) Upon receiving (n_2, R, s, N_1, ID_H, ID_V) in Step 3.a, VLR will verify whether the signature is valid or not. If it holds, VLR will believe that N_1 is



On-line authentication process
MS
VLR
HLR
(σ, K, t, R, s_{Temp}, s_{Next}, K_{new})
(v, K_{VH})
((x, v), (σ, K), K_{VH})
(0) Pre-compute and store
$h^1(n_1), h^2(n_1), ..., h^{n+1}(n_1)$
$K_{new} \equiv_p K^{h(N_1 \| \sigma)}, r \equiv_p g^t$
if $s_{Temp} = NULL, s_{Next} \equiv_q (\sigma * K^{-1} * h(N_1 \| \sigma)^{-1} * K_{new})$
else $s_{Next} \equiv_q (s_{Temp} * K^{-1} * h(N_1 \| \sigma)^{-1} * K_{new})$
(1) K
(2) n_2, ID_V
(3.a)
If (s_{Temp} == Null)
$s \equiv_q \sigma$ +h(N_1||n_2||ID_V)+r*t;
else $s \equiv_q s_{Temp}$+h(N_1||n_2||ID_V)+r*t
(3.b) n_2, R, s, N_1, ID_H, ID_V
(4.a) Check $g^{h(N_1 \| n_2 \| ID_V)} \equiv_p (v^K K^s) r^r$
(4.b) $[N_1 \| n_2 \| K]_{K_{VH}}$, ID_H, ID_V
(5.a) $K_{new} \equiv_p K^{h(N_1 \| \sigma)}$
$C_1 = h(N_1 \| n_2 \| K_{new} \| \sigma)$
$l = N_1$
(5.b) Replace K with K_{new}
(5.c)$[h(N_1 \| K_{new} \| n_2 \| ID_V \| \sigma) \| n_2 \| l \| C_1]_{K_{VH}}$, ID_H, ID_V
(6.a) Check n_2 and $l = N_1$
(6.b) Store C_1, l
(6.c) h(N_1||K_{new}||n_2||ID_V|| σ), ID_V
(6.d) Check h(N_1||K_{new}||n_2||ID_V|| σ)
(6.e) C_1 = h(N_1||n_2||K_{new}|| σ)
Replace (K, s_{Temp}) with (K_{new}, s_{Next})
ith Off-line authentication process
MS (C_i)
VLR (C_i, l = h^{n-i+2}(n_1))
HLR
$[h^{n-i+1}(n_1)]_{C_i}$
Check h(h^{n-i+1}(n_1)) = l
⇒ update l = h^{n-i+1}(n_1)
C_{i+1} = h(l, C_i) and count i = i +1 < n

Fig. 4. Our proposed protocol

generated by MS and believe MS is a privileged user.

b) Upon receiving [h($N_1 \parallel K_{new} \parallel n_2 \parallel ID_V \parallel \sigma$) $\parallel n_2 \parallel l \parallel C_1]_{K_{VH}}$ in Step 5.c, VLR will verify whether n_2 is the same as before. If it is true, VLR believes MS $\overset{SK}{\longleftrightarrow}$ VLR since the secret key K_{VH} is only shared between VLR and HLR. Note that SK is C_1.

c) Since n_2 is chosen by VLR, VLR believes n_2 is fresh and believes that MS believes MS $\overset{SK}{\longleftrightarrow}$ VLR.

d) Using the same way, upon receiving h($N_1 \parallel K_{new} \parallel n_2 \parallel ID_V \parallel \sigma$) in Step 6.c. MS will verify whether the received digest value is correct by using the secret key σ. If it is true, MS will calculate the session key C_1 = h($N_1 \parallel n_2 \parallel K_{new} \parallel \sigma$) and believe that n_2 is generated by VLR and believe MS $\overset{SK}{\longleftrightarrow}$ VLR.

e) Since t and N_1 are chosen by MS, MS will believe N_1 and K_{new} are fresh and believe that VLR believes MS $\overset{SK}{\longleftrightarrow}$ VLR.

2) **User Privacy.** The pseudonym K is renewed when each service request finished. Based on the difficult of the discrete logarithm problem [9], [10], MS will replace the pseudonym K with K_{new} = $K^{h(N_1 \| \sigma)}$ mod p. Without the knowledge of the secret key σ, no one can derive the relationship between K and K_{new}. Hence, we say that the old identifier and the new identifier is unlinkable.

3) **Non-repudiation.** In the proposed protocol, MS has the ability to generate a different signature pair (r, s) from the authorization of HLR. Since only HLR owns the power to authorize MS from signing the signature (the concept of proxy signature [11]), HLR cannot deny this event when a disputation occurs.

4) **Overcharge Problem.** There are two situations that a valid malicious VLR can launch the overcharge problem to HLR successfully.

a) If VLR has the ability to derive the next identifier K_{new}, VLR can feel free to forge the hashing chain h^1(n_1), .., h^{n+1}(n_1) to trick HLR. The overcharge problem happens. As the security analysis of the ”User Privacy”, based on the difficulty of the discrete logarithm problem, no one can derive the K_{new} except valid MS and HLR. This way is infeasible.

b) We suppose n = 20 to demonstrate the attack. Based on the concept of hashing chain, if MS only requires 10 times service from VLR and VLR adds gradually the counter i until i = 10. VLR cannot derive the rest of hashed values such as h^9(n_1), h^8(n_1), ..., and h^1(n_1) for tricking HLR that MS has visited VLR 20 times

TABLE I
COMPARISON OF THE COMMUNICATION COST

	On-line Communication Cost of MS	
	Receive	Send
Our protocol	256 bits	1440 bits
Lee and Yeh's protocol [6]	512 bits	1824 bits
Lee *et al.*'s protocol [7]	512 bits	1888 bits

due to the properties of one way hash function. This way is also infeasible.

5) **Session Key Security.** We discuss several situations for the security of the session key.

   a) From an eavesdropper point of view, the eavesdropper cannot obtain the verifier h$^{n-i+1}(n_1)$ from $[\mathrm{h}^{n-i+1}(n_1)]_{C_i}$ without knowing the secret key σ and the next identifier K_{new}. It implies that the eavesdropper cannot gain the session key C_i or next session key C_{i+1} = h(l, C_i).

   b) From a valid malicious VLR point of view, without the help of MS from sending the verifier h$^{n-i+1}(n_1)$, VLR also cannot calculate the session key C_i or next session key C_{i+1} = h(l, C_i).

B. Efficiency Analysis

1) Communication Cost: We assume that the length of the identity is 32bits, the output length of the one-way hash function such as MD5 is 128bits, and the output block size of the symmetric cryptosystem is 128bits. If the random number is kept secret, the bit-length is 160; otherwise, the bit-length is 64.

We analyze the communication cost of MS in the on-line authentication process as follows. In Step 1, MS sends the identifier K to VLR. The transferred bit size is 512. In Step 3.b, MS sends (n_2, r, s, N_1, ID_H, ID_V) to VLR. The transferred bit size is 928.

In Step 2, VLR sends (n_2, ID_V) to MS. The received bit size is 96. In Step 6.c, VLR sends (h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ), ID_V) to MS. The received bit size is 160. We compare the related protocols [6], [7] and summarize the result in Table I.

2) Computation Cost: We assume that the modular size of addition, subtraction, multiplication, and inverse operations is in 160bits finite field (mod q) and the modular size of exponential operation is 512bits (mod p). We also assume that the pre-computing phase exists all the compared protocols for giving a fair comparison. We denote that T_H is the time of one hash function operation such as MD5; T_{SYM} is the time of one symmetric en/decrypted operation such as DES; T_{MUL} is the time for one modular multiplication; T_{ADD} is the time for one modular addition operation; T_{INV} is the time for one modular inverse operation; T_{PKC} is the time for one signature/verification operation; and T_{EXP} is the time for one modular exponential operation. Note that we ignore the cost of selecting a random number and replacing the data into the SIM card.

We analyze the computation cost of MS in the pre-computing process as follows. MS selects a random number n_1 and performs $n + 1$ times hash function operations of the value n_1. MS then selects another random number t and computes r = g^t mod p, K_{new} = $K^{h(N_1 \| \sigma)}$ mod p. Finally, MS computes s_{Next}, where if s_{Temp} = NULL, s_{Next} = (σ $*$ K^{-1} $*$ h(N_1 $\|$ $\sigma)^{-1}$ $*$ K_{new}) mod q; otherwise s_{Next} = (s_{Temp} $*$ K^{-1} $*$ h(N_1 $\|$ $\sigma)^{-1}$ $*$ K_{new}) mod q. The computation cost is $(n + 1)T_H$ + 2T_{EXP} + 2T_{INV} + 3T_{MUL}.

We analyze the computation cost of MS in the on-line authentication process as follows. In Step 3.a, MS generates a signature (r, s) for the message h(N_1 $\|$ n_2 $\|$ ID_V), where if s_{Temp} = NULL, then s = (σ + h(N_1 $\|$ n_2 $\|$ ID_V) + $t * r$) mod q; otherwise, s = (s_{Temp} + h(N_1 $\|$ n_2 $\|$ ID_V) + $t * r$) mod q. The computation cost is 1T_H + 2T_{ADD} + 1T_{MUL}. In Steps 6.d and 6.e, MS verifies the digest value is the same as h(N_1 $\|$ K_{new} $\|$ n_2 $\|$ ID_V $\|$ σ) and calculates the session key C_1 = h(N_1 $\|$ n_2 $\|$ K_{new} $\|$ σ). The computation cost is 2T_H.

As mentioned in [12], [13], we learn a relationship as follows: 1T_{PKC} $\simeq$ $\frac{5}{3}$ T_{EXP}, 1T_{EXP} $\simeq$ 240T_{MUL}, 1T_{EXP} $\simeq$ 600T_H and the speed of en/decryption operations of the secret-key system is roughly 100 times faster than the signature/verification of the public-key cryptosystem. Finally, we show the compared results in Table II.

By the above comparisons, we also use Table III to show the satisfaction of the requirements between our proposed protocol and the related protocols.

TABLE II
COMPARISON OF THE COMPUTATION COST

	Computation Cost of MS in n = 10		
	Pre-computing	On-line	Off-line
Our protocol	$2T_{EXP}$ + $2T_{INV}$ + $3T_{MUL}$ + $12T_{H}$ $\simeq$ 487.8T_{MUL} + 2T_{INV}	$3T_{H}$ + $2T_{ADD}$ + $1T_{MUL}$ $\simeq$ 2.2T_{MUL} + 2T_{ADD}	10*T_{SYM} $\simeq$ 40T_{MUL}
Lee-Yeh [6]	$1T_{EXP}$ + $1T_{MUL}$ $\simeq$ 241T_{MUL}	$2T_{H}$ + $1T_{ADD}$ + $1T_{MUL}$ + $1T_{SYM}$ $\simeq$ 5.8T_{MUL} + 1T_{ADD}	10*($3T_{H}$ + $1T_{SYM}$) $\simeq$ 52T_{MUL}
Lee *et al.* [7]	$1T_{EXP}$ + $1T_{MUL}$ + $11T_{H}$ $\simeq$ 245.4T_{MUL}	$2T_{H}$ + $1T_{ADD}$ + $1T_{MUL}$ + $1T_{SYM}$ $\simeq$ 5.8T_{MUL} + 1T_{ADD}	10*T_{SYM} $\simeq$ 40T_{MUL}

TABLE III
COMPARISON OF THE REQUIREMENTS BETWEEN OUR PROTOCOL AND THE RELATED PROTOCOLS

	Our protocol	Lee-Yeh's protocol [6]	Lee *et al.*'s protocol [7]
Mutual authentication	Yes	Yes	Yes
User privacy	Yes	No	No
No overcharge problem	Yes	No[1]	No
Non-repudiation	Yes	Yes	Yes
Secure session key	Yes	Yes	Yes
Communication cost	Low	Middle	Middle
Computation cost	Low	Low	Low

1: Lee *et al.* had proven that the overcharge problem exists in the protocol [7].

V. CONCLUSIONS

We have reviewed the previous delegation-based authentication protocols for use in portable communication systems and have pointed out that their protocols do not provide the privacy of mobile users actually. We also have shown that Lee *et al.*'s protocol suffers from the over charge problem. At the same time, we have proposed an improved method. In which, the privacy of mobile users can be protected actually, the over charge problem can be solved, and the on-line communication and computation cost is still low. Finally, in our proposed protocol, the mobile user can dynamically change his delegated signature without registering to HLR again.

A. Acknowledgements

This work is supported in part by National Science Council under the Grant NSC 96-2628-E-002-182-MY3, NSC 98-2221-E-327-026, and Taiwan Information Security Center (TWISC), NSC 98-2219-E-011-001.

REFERENCES

[1] W. Baocang and H. Yupu, "Public key cryptosystem based on two cryptographic assumptions," *IEE Proceedings - Communications*, vol. 152, no. 6, pp. 861–865, 2005.

[2] M.-S. Hwang, C.-C. Chang, and K.-F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *cacm*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[4] Advanced Encryption Standard, http://csrc.nist.gov/encryption/aes/.

[5] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Communications Magazine*, vol. 31, no. 4, pp. 92–100, 1993.

[6] W.-B. Lee and C.-K. Yeh, "A new delegation-based authentication protocol for use in portable communication systems," *IEEE Transactions on Wireless Communications*, vol. 4, no. 1, pp. 57–64, 2005.

[7] T.-F. Lee, S.-H. Chang, T. Hwang, and S.-K. Chong, "Enhanced delegation-based authentication protocol for PCSs," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2166–2171, 2009.

[8] L. Harn and Y. Lin, "A non-repudiation metering scheme," *IEEE Communication Letters*, vol. 5, no. 12, pp. 486–487, 2001.

[9] J. van der Merwe, D. Dawoud, and S. McDonald, "A fully distributed proactively secure threshold-multisignature scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 562–575, 2007.

[10] H. Wang, F. Zhang, and Y. Sun, "Cryptanalysis of a generalized ring signature scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 2, pp. 149–151, 2009.

[11] T. Cao, D. Lin, and R. Xue, "ID-based designated-verifier proxy signatures," *IEE Proceedings-Communications*, vol. 152, no. 6, pp. 989–994, 2005.

[12] RSA Laboratories' Frequently Asked Questions About Today's Cryptography, V4.0[Online], "Available: http://www.rsasecurity.com/rsalabs/faq/."

[13] B. Schneier, *Applied cryptography, 2nd edition*. John Wiley & Sons Inc., 1996.