A Dynamic Mobile Services Access and Payment Platform with Reusable Tickets for Mobile Communication Networks

Hao-Chuan Tsai[1] and Hui-Fuang Ng[2]
[1]Department of Information Technology and Management, Tzu Chi College of Technology, Taiwan, R.O.C.
E-mail: ss241@tccn.edu.tw Tel: +886-3-8572158
[2]Department of Computer Science, University Tunku Abdul Rahman, Malaysia,
E-mail: pangng1@gmail.com Tel: +60-5-46888

***Abstract*—The next generation mobile systems provide high speed data transferring rate to mobile devices. And it is crucial to integrate together two critical issues, the authentication and roaming in heterogeneous networks. For this, Lei et al. proposed the mobile services access and payment mechanism for the next generation mobile systems recently. Their scheme provides a lightweight service access mechanism in which the computation complexity is low on mobile devices. Unfortunately, we found that their scheme suffers from the personal privacy problem. In this paper, we propose an improved version which is lightweight, practical, and likely to avoid re-initialization. When a user roams in heterogeneous networks, he can utilize the delegation reusable ticket to achieve mutual authentication and non-repudiation. Our proposed scheme provides the better efficiency for users in mobile systems.**

I. INTRODUCTION

Wireless mobile networks, such as WiMax [4] and UMTS [2], are attractive and have grown significantly in the last decade. It is highly desirable feature in the development of computer networks and telecommunication systems, especially in wireless networks. Mobile users can access services without geographical limitations through these telecommunication systems, such as cellular networks [1], [11]. To provide such service, a service provider must authenticate mobile users who originally subscribed to their own home networks. However, the traditional access schemes require the exchange of authentication information between the home network and the service provider using roaming agreements [6, 7, 8, 13, 15, 16]. This strategy involves complicated activities over large scale mobile networks.

A ticket is a piece of data which provides a mobile user who can be authenticated by the service provider to access resources. Ticket-based service access is a popular approach in mobile communications where users roam in foreign networks which provide services [9, 10, 12, 14]. Recently, Lei et al.'s proposed a new ticket-based mobile services access scheme [10]. Unlike the other ticket-based schemes, their scheme has two main advantages. The first advantage is that their scheme is lightweight on the mobile user side. Most high computational complexity is progressed in the powerful server side. The second advantage is that the ticket is reusable. They apply two hash chains, the authentication chain and the payment chain, to achieve mutual authentication, non-repudiation, and establishing a session key.

Nowadays, the concern of user privacy is particularly signified in systems which support remote service when users enable to access networks administered though different operators, such desktops and mobile devices. It is highly desirable to provide users anonymous from eavesdroppers as well as the service providers. Unfortunately, we found that Lei et al's scheme still suffers from the user privacy problem. To ready this drawback, we propose a new version which provides both computational efficiency and communication efficiency.

The rest of this paper is organized as follows. We briefly review Lei et al's scheme in Section 2 and demonstrate the weaknesses of their scheme. The enhanced scheme and some discussions are given in Section 3 and Section 4, respectively. Finally, we conclude in Section 5.

II. RELATED WORK

Before demonstrating our concrete scheme, in this section, we first briefly review Lei et al.'s scheme. Next, we will show that there still exist some drawbacks on Lei et al.'s scheme.

A. Briefly Review of Lei et al.'s Scheme

Recently, Lei et al.'s [10] proposed a ticket based mobile service mechanism. Compared with the traditional ticket based mechanisms, such as Kerberos [12], Lei et al.'s scheme eliminates the local clock synchronization problem and also provides a lightweight payment. Different from the similar ticket based schemes, especially, the ticket can be reused on Lei et al.'s scheme for specified times. There exist four principals on their scheme: CA is a certificate authority, S is a ticket server, V is a value-added service provider, and U is mobile user.

Before been authenticating by V, U first needs to communicate with CA to obtain the certificate $Cert_U^S$ and his long term secret key K_U shared with the ticket server S. The certificate is in the form:

$$Cert_U^S =< E_S(U, K_U, \text{expire}, \text{Udata}, Sig_{CA}(U, K_U, \text{expire}, \text{Udata})) >$$

The certificate is encrypted with the ticker server's public key. It is worth noting that CA is also responsible for generating symmetric keys for mobile users. Inside the certificate, U is the mobile user's identity and the field U data includes all other relevant information, such as version number, serial number, user ID, issuer ID, and issuer name.

In their proposed model, a mobile user acquires a ticket before a service request. Once the mobile user has his

certificate, he can start his authentication to the ticket server and obtain a ticket. The protocol can be illustrated as follows:

$$U \rightarrow S : Cert_U^S, \{R_1, R_2, \ldots, R_T\}_{K_U}.$$

$$S \rightarrow U : \{R_1 \oplus R_2 \oplus \ldots \oplus R_T, id, ticket, C_1, C_2, \ldots, C_T, K_{UP}\}_{K_U}$$

The protocol starts when U sends a set of random nonces $R_1, R_2, \ldots, R_T$ encrypted by his long term symmetric key and the certificate of U under S. After receiving these messages, the ticket server S decrypts the certificate with its private key then verifies the signature. When S verifies successfully the certificate from U, it discovers the secret key K_U and stores its relevant information such as the expire date. It obtains the nonces and computes the corresponding hash chains. Also, S generates the hash chains based on the nonces received and billing chains $g^{(M)}(C_1), g^{(M)}(C_2), \ldots, g^{(M)}(C_T)$. Then, it creates a ticket and computes the confirmation $\{R_1 \oplus R_2 \oplus \ldots \oplus R_T, id, ticket, C_1, C_2, \ldots, C_T, K_{UP}\}_{K_U}$. After that, S sends them back to U. Note that the ticket is defined as:

$$ticket = < E_V(V, K_{UV}, id, f^{(N)}(R_1), f^{(N)}(R_2), \ldots, f^{(N)}(R_T), g^{(M)}(C_1), g^{(M)}(C_2), \ldots, g^{(M)}(C_T), Tdata, Sig_S(V, K_{UV}, id, f^{(N)}(R_1), f^{(N)}(R_2), \ldots, f^{(N)}(R_T), g^{(M)}(C_1), g^{(M)}(C_2), \ldots, g^{(M)}(C_T), Tdata)) >.$$

The ticket is specific to the value-added service provider V. When the ticket is sent, simultaneously, the ticket server stores the information pertaining to the ticket for future billing verifications.

Once U obtains a valid ticket, he can grant the service provision from V. The service provision protocol is illustrated as follows (for ith authentication):

$$U \rightarrow V : ticket, \{f^{(N-i)}(R_t), V, N_U\}_{K_{UV}}.$$

$$V \rightarrow U : \{f^{(N-i+1)}(R_t), N_U + 1, N_V, T_V\}_{K_{UV}}$$

$$U \rightarrow V : \{N_V \oplus T_V\}_K.$$

Initially, U sends the ticket, along with the hash chain i ($1 \le i \le N$ *and* $1 \le t \le T$), the identity of V and a nonce N_U encrypted with the authentication key K_{UV} previously generated by the ticket server S. When V receives the service request, it decrypts the ticket with its private key and verifies the signature of S. Next, it compares this signature to the one received. If they are identical, the ticket is considered as valid. Then, V retrieves the stored hash chain $f^{(N-i+1)}(R_t)$ and verifies if the hash chain $f^{(N-i)}(R_t)$ satisfies the equation:

$$f(f^{(N-i)}(R_t)) = f^{(N-i+1)}(R_t).$$

If so, it updates the hash chain by removing the current tip $f^{(N-i)}(R_t)$. After that, the ith hash chain is abandoned and the (i+1)th hash chain is used until all the hash chains run out.

Next, V chooses a random nonce N_V and the current timestamp T_V and encrypts the confirmation messages $\{f^{(N-i+1)}(R_t), N_U + 1, N_V, T_V\}$ by using the authentication key K_{UV}. Simultaneously, V also computes the fresh session key $K = f(N_U \oplus N_V)$. When U receives the messages, he decrypts the messages and verifies them with the same procedure. If they hold, U authenticates V successfully and he also computes the fresh session key $K = f(N_U \oplus N_V)$ and sends back $\{N_P \oplus T_V\}_K$ to V for mutual authentication.

B. The Disadvantages of Lei et al.'s Scheme

Although Lei et al. proposed an efficient mobile service access scheme with reusable tickets, there still does not satisfy some basic security requirements. We now describe the disadvantages found in Lei et al.'s scheme.

The problem is the personal privacy problem. The concern of user privacy is particularly signified in systems which support remote service when users enable to access networks administered though different operators, such desktops and mobile devices. It is highly desirable to provide users anonymous from eavesdroppers as well as the service providers. Unless the identity information becomes critical in some emergency situation or special applications, only the trusted certificate authority can recover the truly identities of users, for example, one of such applications is pay-TV systems. Ideally, a user should reveal nothing to the service provider other than the confirmation of his ordered channels with respect to his real identity. The other important issue is user untraceability. This property ensures that eavesdroppers cannot identify two transactions which are corresponding to the same mobile user or not. Unfortunately, Lei et al.'s scheme can be traceable to eavesdroppers easily for the specific mobile user. This is mainly due to a reason, the used ticket is unchangeable. For granting the service from the service provider, a mobile user needs to be authenticated through the ticket generated by the ticket server. Even such the ticket involves encryption and digital signature from the ticket server to prevent the ticket duplication problem, nevertheless, the encrypted messages are fixed rather than dynamic, and it is easy for eavesdroppers to trace two different transactions with the same ticket to the specific mobile user. Consequently, the untraceability between two different transactions cannot be achieved in Lei et al.'s scheme.

III. THE PROPOSED SCHEME

Before demonstrating the concrete scheme, in this section, we first give the formal description of the application communication model, followed by the introduction of the bilinear pairings [3], which are the foundation of the proposed scheme.

A. Preliminaries

Let G_1 and G_2 be two cyclic groups of order p for some large prime p and be equipped with pairings. We suppose that an efficient computable bilinear map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ must satisfy the following properties:

1. Bilinear: a map $\hat{e} : G_1 \times G_2 \rightarrow G_T$ is said bilinear if $\hat{e}(\alpha P, \beta Q) = \hat{e}(P, Q)^{\alpha\beta}$, for all $P \in G_1, Q \in G_2$ and all $\alpha, \beta \in Z_p^*$.

2. Non-degenerate: the map does not send all pairs in $G_1 \times G_2$ to the identity in G_T. This property implies that if P is a generator in G_1 and Q is a generator in G_2, then $\hat{e}(P,Q)$ is also a generator in G_T.
3. Computable: there exists an efficient algorithm to compute $\hat{e}(P,Q)$ for any $P \in G_1, Q \in G_2$.

For the sake of convenience, most of the researches also apply in the symmetric setting, where $G_1 = G_2$. In this paper, we also apply the symmetric setting to the proposed scheme. Next, we depict the security basis on which our scheme relies.

DEFINATION. Bilinear Deffie Hellman Assumption (BDH): Let G_1 and G_T be two groups with prime order *p*. Let $\hat{e}: G_1 \times G_1 \to G_T$ be a bilinear map and let P be a generator of G_1. Given $< P, \alpha P, \beta P, \gamma P >$ for some $\alpha, \beta, \gamma \in Z_p^*$ compute $W = \hat{e}(P,P)^{\alpha\beta\gamma} \in G_T$. For unknown $\alpha, \beta, \gamma \in Z_p^*$ and $W = \hat{e}(P,P)^{\alpha\beta\gamma} \in G_T$, we say that an algorithm A has an advantage $\varepsilon(k)$ in solving the above problem for sufficiently large *k*:

$$Adv(A) = \Pr[A(G_1, G_T, \hat{e}, \alpha P, \beta P, \gamma P, \hat{e}(P,P)^{\alpha\beta\gamma}) \mid P \in G_1^*, \alpha, \beta, \gamma \in Z_p^*] - \Pr[A(\alpha P, \beta P, \gamma P, W) \mid \alpha P, \beta P, \gamma P \in G_1^*, W \in G_T] > \varepsilon(k).$$

We say BDH assumption holds if any randomized polynomial time algorithm A solves the above problem with advantage with at most $\frac{1}{f(k)}$ for any polynomial $f \in Z[x]$. Informally speaking, given a random tuple $< G_1, G_T, \hat{e} >$ generated by the systems, there exists no efficient algorithm that can solve BDH problem in $< G_1, G_T, \hat{e} >$ with non-negligible advantage.

B. The Concrete Scheme

Herein, we present the mobile access platform with the general trading model. We reasonable assume that the identities of the home domain server and service provider can be easily granted for mobile users. The design principal of the proposed scheme is that: to grant the service, a mobile user U_a should acquire a valid ticket from the corresponding home domain server HS_h. Similar to *Lei et al.*'s scheme, we also preserve the merit of reusable tickets to reduce the communications and bandwidth with HS_h.

Firstly, the system initializes the security parameters which are described as follows. Firstly, the system chooses a finite field F_p over a large odd prime number *p*, and then, defines an elliptic curve equation $E_p(a,b): y^2 = x^3 + ax + b \pmod p$ with the prime order q over F_p, for the chosen $a, b \in F_p$ satisfying the equation $4a^3 + 27b^2 \neq 0 \pmod p$ [5]. Finally, the system chooses a base point *P* with the prime order q over $E_p(a,b)$ and publishes $E_p(a,b)$ and *P*. In addition, the system selects three cryptographic hash functions $H_1: G_T \to \{0,1\}^{l_1}$, $H_2: \{0,1\}^* \to \{0,1\}^{l_2}$, $H_3: \{0,1\}^* \to \{0,1\}^{l_3}$, and $H_4: \{0,1\}^* \to \{0,1\}^{l_4}$ for the length l_1, l_2, l_3 and l_4. Now, we present the concrete scheme as follows.

Mobile user registration phase

Before granting the service of the service provider, the mobile user U_a only sends his real identity ID_{U_a} to the specific home domain server once time. To provide user's anonymity and future charging, the specific home domain, named HS_a, must compute the temporary but relative identity to the real identity ID_{U_a} for the mobile user U_a.

Firstly, HS_a chooses a private key $x_{HS_a} \in Z_q^*$ and computes the corresponding public key $Y_{HS_a} = (x_{HS_a} + H_3(ID_{HS_a})) \cdot P$. And then, for the mobile user U_a, HS_a selects the master delegation secret δ_{U_a} which satisfies the following equation:

$$H_3(x_{HS_a}, ID_{HS_a}, ID_{U_a}) = H_3(ID_{U_a}, ID_{HS_a}) \cdot \delta_{U_a} \mod q.$$

Next, HS_a computes $\gamma_{U_a} = \delta_{U_a} \cdot P = (x_{U_a}, y_{U_a})$ and $c = x_{U_a} \bmod q$, simultaneously, HS_a also calculates equations as follows:

$$e_{U_a} = H_3(c, ID_{HS_a}), \text{ and}$$
$$s_{U_a} = \delta_{U_a} - x_{HS_a} * e_{U_a} \mod q.$$

Eventually, HS_a computes $H_3(\delta_{U_a} \| ID_{HS_a})$ as the master delegation key and delivers the computed result along with (e_{U_a}, s_{U_a}) to the mobile user U_a securely. It is worth noting that HS_a destroys all the secret information but only his private key x_{S_h}. That is, we assume that the private key x_{S_h} is under the restrict protection.

In addition, U_a can utilize Y_{HS_a}, the public key of HS_a, to ensure the integrity of the received secrets. The received secrets can be verified successfully since

$$\begin{aligned}\gamma'_{U_a} &= s_{U_a} * P + Y_{HS_a} * e_{U_a} \\ &= s_{U_a} * P + x_{HS_a} * P * e_{U_a} \\ &= (s_{U_a} + x_{HS_a} * e_{U_a}) * P \\ &= (\delta_{U_a} - x_{HS_a} * e_{U_a} + x_{HS_a} * e_{U_a})P = \delta_{U_a} * P = \gamma_{U_a}.\end{aligned}$$

Ticket acquisition phase

Before granting the service providers, the mobile user U_a has to obtain the valid ticket from the ticket server which is integrated with the corresponding home server HS_a. The

design principle of this procedure utilizes the reusable ticket to eliminate the communication overhead for the multi-server architecture. The message flow for this phase is the following steps:

Step T1: U_a first generates a set of random elements $\{\alpha_1, \alpha_2, \ldots, \alpha_T\}$ and encrypts them with the identity of the home server by the master delegation key $H_3(\delta_{U_a} \| ID_{HS_a})$ and then sends the encrypted result $E_{H_3(\delta_{U_a}\|ID_{HS_a})}[\alpha_1, \alpha_2, \ldots, \alpha_T, ID_{HS_a}]$ along with the certificate-like messages (e_{U_a}, s_{U_a}) to the ticket server.

Step T2: After receiving the messages from U_a, on this stage, the ticket server initially utilizes the certificate-like messages (e_{U_a}, s_{U_a}) and his private key x_{HS_a} to compute $\delta'_{U_a} = s_{U_a} + x_{HS_a} * e_{U_a} \bmod q$, and verifies the computed result between δ'_{U_a} and e_{U_a} with the aforementioned steps in the mobile user registration phase. If the verification succeeds, the ticket server recovers the master delegation key $H_3(\delta'_{U_a} \| ID_{HS_a})$ and decrypts $E_{H_3(\delta_{U_a}\|ID_{HS_a})}[\alpha_1, \alpha_2, \ldots, \alpha_T, ID_{HS_a}]$ to retrieve the random elements which are used to be future authentications. Next, for future charging, the ticket server also generates a set of random elements $\{\beta_1, \beta_2, \ldots, \beta_T\}$ and then the ticket server generates a valid and unforgeable ticket which is defined as:

$$ticket = Sig_{x_{HS_a}} < Ticket\_data, H_3(H_2^N(\alpha_1) \oplus H_2^M(\beta_1)), H_3(H_2^N(\alpha_2) \oplus H_2^M(\beta_2)), \ldots, H_3(H_2^N(\alpha_T) \oplus H_2^M(\beta_T)), H_3^2(e_{U_a}) >,$$

where $Sig_{x_{HS_a}} < m >$ denotes that the message *m* is signed by the ticket server's private key x_{HS_a}. It is worth noting that *Ticket_data* field includes all detailed ticket information, such as issuer *ID*, issuer name, issued time, and times of access. In order to be utilized for multiple service providers, the ticket adopts the hash function to protect the authentication and payment information rather than encrypts with public keys of service providers.

Eventually, the ticker server utilizes the recovered master delegation key to encrypt $E_{H_3(\delta_{U_a}\|ID_{HS_a})}[\{\beta_1, \beta_2, \ldots, \beta_T\}, ID_{HS_a}, N, M, ticket]$ and then sends the encrypted result to the mobile user U_a.

Step T3: Upon receiving the ticket server's response, U_a utilizes the master delegation key to retrieve the received result. At first, U_a randomly chooses one of the retrieved $\{\beta_1, \beta_2, \ldots, \beta_T\}$, without loss of generality, we assume that U_a chooses β_1, and then U_a utilizes the previous generated α_1, the retrieved *N* and *M*, and the chose β_1 to compute $H_3(H_2^N(\alpha_1) \oplus H_2^M(\beta_1))$. Next, U_a verifies whether the computed result equals the corresponding field in the ticket. If it holds, the ticket is valid and can be utilized to grant the service.

Service access phase

To provide the service, every service provider has to public his ephemeral key pair. Similar to the ticket server, the service provider P_v chooses a random number $x_{P_v} \in Z_q^*$ as the master secret. In addition, P_v publishes the identity function $F(ID_{P_v}) = (x_{P_v} + H_3(ID_{P_v})) \cdot P$.

Once U_a obtains a valid ticket from the ticket server, then he can use it to grant the service from the service provider P_v. The message flow for this phase is the following steps:

Step S1: U_a chooses a random number $a \in Z_q^*$ as an ephemeral secret to compute rP and generates an encapsulated key $Encaps_K = e(P, P)^a$. Simultaneously, U_a computes $c = a \cdot F(ID_{P_v})$ and $H_1(e(P, P)^a)$, and utilizes the ticket information to compute the following messages

$$ticket \oplus H_4(e(P,P)^a), c, \{H_2^{N-i}(\alpha_t) \oplus H_1(e(P,P)^a) \oplus H_3(e_{U_a})\},$$

$$E_{H_1(e(P,P)^r)}[H_2^{M-j}(\beta_t), ID_{P_v}, ap, e_{U_a}, ts],$$

where *ts* is the timestamp and $1 \le i \le N, 0 \le j \le M, 1 \le t \le T$. Next, U_a sends these messages to the service provider P_v.

Step S2: When P_v receives the service request, he will utilize his master secret to derive $usk_{ID} = \frac{1}{x_{P_v} + H_2(ID_{P_v})} \cdot P$ and to compute the corresponding encapsulated key

$$\begin{aligned} Encaps_K &= e(usk_{ID}, c) \\ &= e(\frac{1}{x_{P_v} + H_2(ID_{P_v})} \cdot P, a(x_{P_v} + H_2(ID_{P_v})) \cdot P) \\ &= e(P, P)^a. \end{aligned}$$

Next, P_v performs the H_1 hash function operation on the derived encapsulated key to retrieve the ticket, also, P_v decrypts the messages $\{H_2^{M-j}(\beta_t), ID_{P_v}, ap, e_{U_a}, ts\}$ from the fourth encrypted item. Then, P_v at first verifies whether the identity ID_{P_v} and timestamp *ts* are valid or not. If they hold, P_v utilizes the decrypted e_{U_a} to retrieve the authentication token $H_2^{N-i}(\alpha_t)$. If the ticket is used for the first time, that means *i*=1 and *j*=0, and P_v computes $H_3(H_2(H_2^{N-i}(\alpha_t)) \oplus H_2^{M-j}(\beta_t))$. Then, P_v compares the computed result with the information in the ticket, which is signed by the ticket server. If the verification succeeds, P_v stores the authentication tokens in the form of

$H_2^N(\alpha_t) \oplus H_2(x_{P_v} \| e_{U_a})$ and $H_3^2(e_{U_a})$; otherwise, P_v checks if the hash chain $H_2^{N-i}(\alpha_t)$ satisfies the equation $H_2(H_2^{N-i}(\alpha_t)) = H_2^{N-i+1}(\alpha_t)$.

Simultaneously, P_v chooses a random number $b \in Z_q^*$ to compute bP and then to generate $H_3(H_2^{N-i+1}(\alpha_t) \| ts) \oplus bP$. After that, P_v computes a fresh session key $sk_{PU} = H_2(abP \| ts \| ID_{P_v} \| e_{U_a})$, and for future charging and tracing, P_v generates a transaction number TN_i and encrypts $\{TN_i, aP, bP\}$ by using the fresh session key. Eventually, P_v sends the encrypted result along with $H_3(H_2^{N-i+1}(\alpha_t) \| ts) \oplus bP$ to U_a.

Step S3: Upon receiving the messages from P_v , U_a first retrieves bP from the first item of the received message $H_3(H_2^{N-i+1}(\alpha_t) \| ts) \oplus bP$ and then utilizes the previously generated random element *a* and the timestamp *ts* to compute the fresh session key $sk_{UP} = H_2(abP \| ts \| ID_{P_v} \| e_{U_a})$. Finally, U_a utilizes the derived session key to decrypt the second item of the received message and to verify the validity of the decrypted $\{aP, bP, TN_i\}$. If the decrypted results equal the previously generated *aP* and the retrieved *bP*, then U_a authenticates P_v successfully. In addition, U_a computes $Auth_U = H_3(ID_{P_v} \| aP \| bP \| TN_i)$ and then sends the computed result to P_v.

Step S4: When P_v receives the message, he also utilizes the previously decrypted *aP*, the identity of P_v , and the generated *bP* , TN_i to perform the $Auth_P = H_3(ID_{P_v} \| aP \| bP \| TN_i)$. If the computed $Auth_P$ equals the received $Auth_U$, the P_v authenticates U_a successfully, and P_v can provide the service to U_a with the secure communications by using the established session key $H_2(abP \| ts \| ID_{P_v} \| e_{U_a})$.

IV. DISCUSSIONS

We discuss the security of the proposed scheme by considering the possible attacks as in the following:

(1). Mutual authentication: initially, the service provider and the mobile user can achieve mutual authentication through the certificate-like message (e_{U_a}, s_{U_a}) . During the service access phase, the ticket is protected by the service provider's public key, if the ticket is valid, the service provider can verify it to authenticate the mobile user and vice versa.

(2). Forge tickets: it is computational infeasible to forge tickets, because tickets are protected by service providers' public keys.

(3) Impersonation attacks: if an adversary wants to impersonate the service provider to cheat a mobile user, he must produce the responding confirmation (e_{U_a}, s_{U_a}) to generate the valid ticket, which implies that an adversary processes the long-term secret key. In addition, to be verified by a service provider, an adversary still must compute the master delegation key $H_3(\delta'_{U_a} \| ID_{HS_a})$ to derive the legitimate ticket. Hence, an adversary cannot impersonate both the service provider and the mobile user.

(4). Renewal session key: the established session key $H_2(abP \| ts \| ID_{P_v} \| e_{U_a})$ is used only during the authentication by the service provider and the mobile user. The session key is generated in each authentication phase. Both the service provider and the mobile user contribute the random nonces to establish the session key.

V. CONCLUSIONS

In this paper, we have demonstrated that Lei et al's scheme has still vulnerable to several weaknesses. To remedy these weaknesses, we have proposed an improved scheme with the novel and efficient architecture. We focus on preserving the secrecy of mobile users' identities for large scale wireless networks. The proposed scheme has several attractive characteristics as follows, i.e., the mutual authentication; the home server does not have to store the long-term keys shared with mobile users which is scalable in wireless networks; the privacy of mobile users' information is maintained, and the established session key can only be shared by the communication parties, and, even the home server cannot obtain the established session key. In addition, it is well-suited for the low power devices used with wireless networks.

REFERENCES

[1] 3GPP, "Wireless Local Area Network (WLAN) Interworking Security," *3GPP TS 33.234*, 2004.

[2] 3GPP, "Universal Mobile Telecommunication System," *3GPP TS 25.101*, 2004.

[3] P. Barreto, L. Ben, and S. Michael, "Efficient Implementation of Pairing-based Cryptosystems," *Journal of Cryptology*, vol. 17, no. 4, pp. 321–334, 2004.

[4] M. Ergen, *Mobile Broadband - Including WiMAX and LTE*, Springer, NY, 2009.

[5] D. Hankerson, A. Menzes, and S. Vanstone, Guide to Elliptic Curve Cryptography, Second Edition, Springer-Verlag, New York, USA, 2004.

[6] L. Harn and W. Hsin, "On the Security of Wireless Network Access with Enhancements," *Proceedings of the 2003 ACM workshop on Wireless Security*, San Diego, USA, pp. 88–95, Sept. 2003.

[7] C. Huang and J. Li, "Authentication and Key Agreement Protocol for UMTS with Low Bandwidth Consumption," *Proceedings of the 19th International Conference on Advanced*

Information Networking and Applications (*AINA* 2005), Taipei, Taiwan, IEEE Computer Society, pp. 392–397, Mar. 2005,

[8] C.H. Lee, M.S. Hwang, and W.P. Yang, "Enhanced Privacy and Authentication for the Global System for Mobile Communications," *Wireless Networks*, vol. 5, pp. 231–243, 1999.

[9] B. Lee, T. Kim and S. Kan, "Ticket Based Authentication and Payment Protocol for Mobile Telecommunications Systems," *Proceedings of the Pacific Rim International Symposium, Dependable Computing*, pp. 218–221, Dec. 2001.

[10] Y. Lei, A. Quitero, and S. Pierre, "Mobile service access and payment through reusable tickets," *Computer Communications*, vol. 32, pp. 602–609, 2009.

[11] R. Molva, D. Samfat, and G. Tsudik, "Authentication of mobile users," *IEEE Network*, *Special Issue on Mobile Communications*, vol. 8, no. 2, pp. 26–34, 1994.

[12] B.C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks," *IEEE Communications*, vol. 32, no. 9, pp. 33–38, 1994.

[13] M. Rahnema, "Overview of the GSM System and Protocol," *IEEE Communications Magazine*, vol. 31, pp. 92–100, 1993.

[14] H. Wang, J. Cao and Y. Zhang, "Ticket-based Service Access Scheme for Mobile Users," Australian Computer Science Communications, vol. 24, no. 1, pp. 285–292, 2002.

[15] H. Wang, Y. Zhang, J. Cao and V. Varadharajan, "Achieving Secure and Flexible Services through Tickets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol.33, no. 6, pp. 697–708, 2003.

[16] M. Zhang and Y. Fang, "Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 734–742, 2005.