An Enhanced Secure Delegation-based Anonymous Authentication Protocol for PCSs

Prosanta Gope*
National University of Singapore, Singapore

Hemant Ghayvat
Technical University of Denmark, Denmark

Yongqiang Cheng
University of Hull, UK

Sohag Kabir
University of Bradford, Bradford, UK

Summary — **Rapid development of wireless networks brings about many security problems in Portable Communication Systems (PCSs), which can provide mobile users with an opportunity to enjoy global roaming services. In this regard, designing a secure user authentication scheme, especially for recognizing legal roaming users is indeed a challenging task. It is noticed that, there is no delegation-based protocol for PCSs, which can guarantee anonymity, untraceability, perfect forward secrecy and resistance of Denial of Service (DoS) attack. Therefore, in this article we put forward a novel delegation-based anonymous and untraceable authentication protocol, which can guarantee to resolve all the above mentioned security issues and hence offer a solution for secure communications for PCSs.**

Index Terms — Privacy, Anonymity, Untraceability, Authentication, Smartcard, Global mobility networks.

1. INTRODUCTION

Portable Communication Systems (PCSs) offer roaming services among wireless communication networks [1-4]. In this regard, a mobile user (MU) at first registers his/her legality into a home location register (HLR). Before roaming, MU logins the visiting location register (VLR) and VLR validates the user's legality with the help of the registered HLR. If the MU is legal, VLR offers services and charges the roaming fee. In this context, an adversary can easily perform more and more malicious activities with the existing PCSs by capturing the content of communication without making any physical connections to the wireless network. Therefore, the security issue has always been a major concern in wireless networks. It is the general trend of the earlier authentication protocols to use the costly public-key cryptosystem systems (PKCs) to offer security and privacy to the mobile users [5-10]. In 2005, Based on the PKC, W.B.Lee and Yeh [5] put forward a delegation-based identification scheme for PCSs, which provides user anonymity. The protocol also ensures low communication cost since it is based on the off-line authentication mechanism. However, T.F. Lee et al. [6] investigated the protocol in [5] and showed that it is insecure against the impersonation attack. Thereafter, T.F.Lee et al. designed a revised protocol in [6] to eliminate the impersonation attack of the protocol in [5]. However, Youn et al. [7], and Wang et al. [8] proved that the protocol in [6] cannot tolerate the linkable problem. In addition to that, C.C.Lee et al. [9] analyzed and argued that T.F.Lee et al.'s protocol [6] cannot overcome the problem of forward secrecy of the session key, which leads to computation of current session key based on the disclosure of a previous session key. C.C. Lee et al. [9] then designed an enhanced protocol to dispense with the existing problems of the protocol in [6]. Unfortunately, in this article we show that the protocol in [9] is not flexible and robust since it is suffering from some security threats including the Denial of Service (DoS) attack, side channel attack, and perfect forward secrecy and untraceability problem. Besides, according to [10], there is no delegation-based authentication protocol, which can guarantee untraceability, perfect forward secrecy, and the resistance of DoS attack. In this article, we put forward an anonymous and untraceable delegation-based authentication process for PCSs, which can guarantee

* Corresponding Author: **Prosanta Gope**

to resolve all the security issues existing in other delegation-based protocols and even can offer a secure and expeditious PCS with the lower computation and communication costs.

The remainder of this article has been organized as follows. In Section 2, we pinpoint the weaknesses of the protocol proposed in [9] . Thereafter, we present our proposed scheme in Section 3, whose security and performance are analyzed in Section 4 and Section 5, respectively. Finally, a concluding remark is given in Section 6. The abbreviations and cryptographic functions used in this article are defined in Table 1.

Table 1 Notations and Abbreviations

Symbol	Definition
MU	Mobile User
VLR	Visiting Location Register
HLR	Home Location Register
K_{mh}	Secret Key between the MU and HLR
K_{vh}	Secret Key between the VLR and HLR
ID_H	Identity of the HLR
ID_V	Identity of the VLR
SK	Session key between VLR and MU
$\boldsymbol{p}$	A large prime
$\boldsymbol{q}$	A prime factor of *p-1*
$\boldsymbol{g}$	A generator in group z_p^*
$[M]E_K$	Encryption of a message M using secret key *K*
$h(.)$	One-way hash function
$\oplus$	Exclusive-OR operation
$\|$	Concatenation operation

2. SECURITY WEAKNESSES IN [9]

In this section, we present the several weaknesses of C.C. Lee et al.'s protocol in [9] (shown in Fig. 1), which certainly cause security concerns in wireless communication system.

2.1. *Vulnerable to DoS Attacks*

DoS attack [10-11] is an imperative concern, which may occur attributed to the loss of synchronization between MU and HA. That can be comprehended if the last authentic response message sent by VLR has been interrupted by an adversary, so that MU cannot receive the message within a specific time period. Unfortunately, C.C. Lee et al.'s protocol cannot resist DoS attack, where if an adversary interrupts the response message (Step 6 in Fig.1), then MU cannot receive $[N_1, n3, ID_V, T_{IDnew}]E_\sigma, ID_V$. In that case, both the HLR and VLR may update their databases with temporary id T_{IDnew} but MU cannot. Now, if the MU attempts to execute the "*On-line Authentication Phase"* with the old temporary id

T_{ID} then the HLR will not comprehend that. On the other hand, because of the interruption of the message in Step 6, MU cannot even acquire the random number *n3* and without *n3*, it is impossible for MU to compute the session key $SK = C_1 = h(N_1 \| n2 \| n3 \| \sigma)$. Therefore, without having session key and the temporary id T_{IDnew}, the MU cannot even execute the "*Off-line Authentication Phase"*.

2.2. *Loss of Untraceability*

An orthogonal security arising as a result of mobility is the privacy of the mobile subscriber's identity and his/her movements. For obvious reasons, it is desirable to keep this information secret. In other words, passive eavesdroppers and active intruders should not be able to identify or keep track of the user. In fact, it can be argued that even the VLRs should not be privy to know any identification information of the user. Unfortunately, in Step 5 of C.C. Lee et al.'s protocol, the VLR receives the MU's latest temporary identity T_{IDnew} from HLR. Now, if the MU moves to a new VLR, the old VLR can still track him/her. In this way, the protocol proposed by C.C. Lee et al. compromises the domain separation [3-4] property, [12] that means conspiracy of all the visited domains may cause to identify the movement of the user. Therefore, C.C. Lee et al.'s protocol cannot ensure the untraceability property, which is greatly important for the privacy of the mobile user.

2.3. *No Perfect Forward Secrecy*

Perfect forward secrecy [13] is a form of security requirements in network systems. In general, a protocol that provides perfect forward secrecy (PFS) can resist an adversary from learning any previous session keys, especially when the long term secret keying material is compromised by the adversary. However, we found that C.C. Lee et al.'s protocol for PCSs fails to provide PFS. In C.C. Lee et al.'s delegation-based protocol, once the secret key pair (K, σ) is disclosed, then all the previous session keys established based on the execution "*On-line Authentication Phase"* will be exposed. Precisely, an adversary can learn the previous session key if the home agent is compromised by the adversary. So that, the adversary may acquire secret key pair (K, σ) and/or the shared secret key K_{vh}. Therefore, the session key in this scheme is not secure. In fact, W.B. Lee and Yeh's scheme [5] and T. F. Lee et al.'s scheme [6] also cannot ensure PFS.

2.4. *Vulnerable to Side Channel Attacks*

In practice, it is possible to read some sensitive information from SIM card by executing the side channel attacks [14], and the information can be used for breaking the whole system. Hence, it is highly desirable to use countermeasures for securing the secret values stored in SIM card. However, sometimes, developers do not use countermeasures due to expensive production cost. In this regard, the best alternative plan is to ensure the security of unspoiled SIM cards by restricting the damage caused by the revelation of sensitive information. Unfortunately, C.C. Lee et al.'s delegation-based protocol can be entirely broken, since an adversary can always recover the key pair (σ, K), the latest temporary identity of the MU i.e. T_{IDnew}, and even the latest hash chain values with the session key C_i from the SIM card. Once the adversary obtains these parameters, then he/she can easily impersonate as MU, which is a serious threat against the privacy of the mobile user. Similar problem can also be found in [5-9].

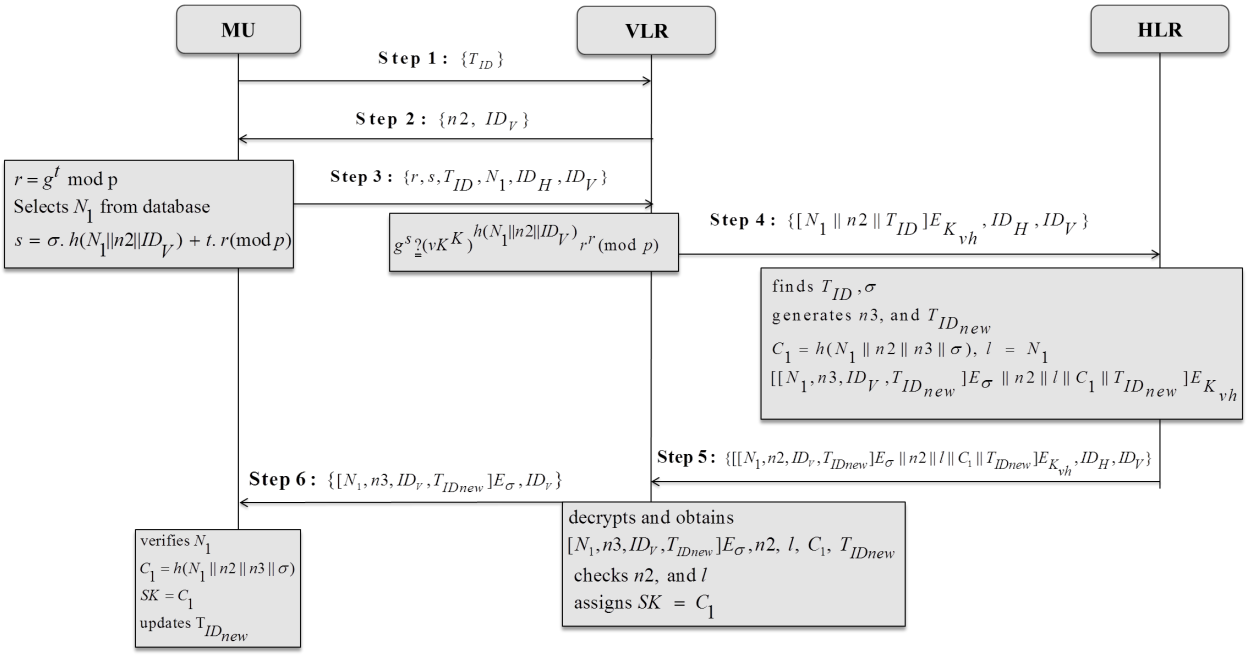

Fig. 1 The On-line Authentication Process of C.C. Lee et al.'s Protocol

3. PROPOSED PROTOCOL

To mitigate the security and functionality issues in the existing solutions, in this article we design a novel authentication protocol for PCSs. The list of notations is explained in Table 1. Our protocol consists of three phases. In **Phase I**, a MU registers to a HLR and then the HLR issues a smartcard for the MU. The **Phase II**, which is an on-line authentication phase, is executed when a MU roams into a foreign network to obtain the ubiquitous services and in this situation, the VLR authenticates the MU with the support of the HLR. The **Phase III**, which is an off-line authentication phase, is executed when a MU roams into a foreign network and the current VLR authenticates the MU without any help from the HLR. The design goals of our protocol can be summarized in the following ways:

- It provides mutual authentication;
- It ensures user anonymity with the resistance of DoS attacks;
- It provides privacy against eavesdropper (PAE);
- It resists different attacks and provides forward/backward secrecy;
- It offers perfect forward secrecy;
- It has low computation and communication cost.

A. ***Phase I****: Registration Phase*

For registration, a new MU submits his/her identity ID_M to a particular HLR using a secure channel. Then the HLR selects a nonce n_h, and calculates $K_{mh} = h(ID_M \| n_h) \oplus ID_H$. Furthermore, the HLR generates a set of unlinkable pseudo-identities and emergency key pairs $(PID, K_{em}) = \{(pid_1, k_{em_1}), (pid_2, k_{em_2}), ..., (pid_n, k_{em_n})\}$ for the corresponding MU. Besides, the HLR maintains a transaction sequence number Ts_{mh} and it is calculated depending on the number of authentication requests (m) handled by the HLR, including the present request of the current MU. For every request, the HLR sets $m \leftarrow m + 1$ and $Ts_{mh} \leftarrow m$ and subsequently sends Ts_{mh} to the MU while maintaining a copy of it in its database. This sequence number will help HLR to identify the consumer. Besides, based on that, the HLR can also take decision on whether a MU's request is valid. If the Ts_{mh} provided by MU is not valid, then HLR instantly aborts the request and the MU will be asked to use his/her one of the unused pair of (pid_j , k_{em_j}). Once a pair is consumed, then that pair must be deleted from the list by both the MU and the HLR. Now, the HLR personalizes a smartcard with $\{K_{mh}, (PID, K_{em}), Ts_{mh}\ h(\cdot)\}$ and delivers it to the MU securely for further communication. The HLR stores both the MU ID and the smartcard, i.e. a copy of $\{ID_M, K_{mh}, (PID, K_{em}), Ts_{mh}\}$ into the database. Upon receiving the smartcard, the MU generates a nonce $n1$, and pre-computes a hash chain $h^{(1)}(n1), h^{(2)}(n1), ..., h^{(n+1)}(n1)$ and stores them in its database, where $h^{(1)}(n1) = h(n1)$ and $h^{(i+1)}(n1) = h(h^{(i)}(n1))$ for i = 1, 2, ...,n. Thereafter, the MU selects a low-entropy password PSW_M and calculates $K^*_{mh} = K_{mh} \oplus h(ID_M \| PSW_M), PID^* = PID \oplus h(ID_M \| PSW_M), K^*_{em} = K_{em} \oplus h(ID_M \| PSW_M)$. Finally, the MU replaces K_{mh} with K^*_{mh}, PID with PID^*, and K_{em} with K^*_{em} then the smartcard finally holds $\{K^*_{mh}, (PID^*, K^*_{em}), Ts_{mh}, h(\cdot)\}$.

B. ***Phase II****: On-line Authentication Phase*

This phase is needed to authenticate a MU by the VLR with the help of HLR when the MU roams into the region of a FN. In this phase, the VLR authenticates the MU and the HLR and the MU mutually computes a session key SK . This phase consists of the following steps:

Step 1 $\mathbf{M_{B_1}} : \text{MU} \rightarrow \text{VLR} : \{AID_M, N_1, Ts_{mh}(if\ req.), V_0, ID_H\}$.

MU at first picks $N_1 = h^{(n+1)}(n1)$ from its database and then keys the smartcard into the device and inputs ID_M and PSW_M. The smartcard then calculates $K_{mh} = K^*_{mh} \oplus h(ID_M \| PSW_M)$. Hereafter, it derives $AID_M = h(ID_M \| K_{mh} \| N_1 \| Ts_{mh})$, $V_0 = h(AID_M \| K_{mh} \| N_1)$. As a final point, the MU delivers the authentication message $\mathrm{M_{B_1}}$ to the VLR over a public channel. If there is a loss of synchronization between the MU and the VLR, the MU selects an unused pair of $(pid^*_j, k^*_{em_j})$ and then inputs ID_M and PSW_M into the smartcard. The smartcard then calculates $pid_j = pid^*_j \oplus h(ID_M \| PSW_M)$,

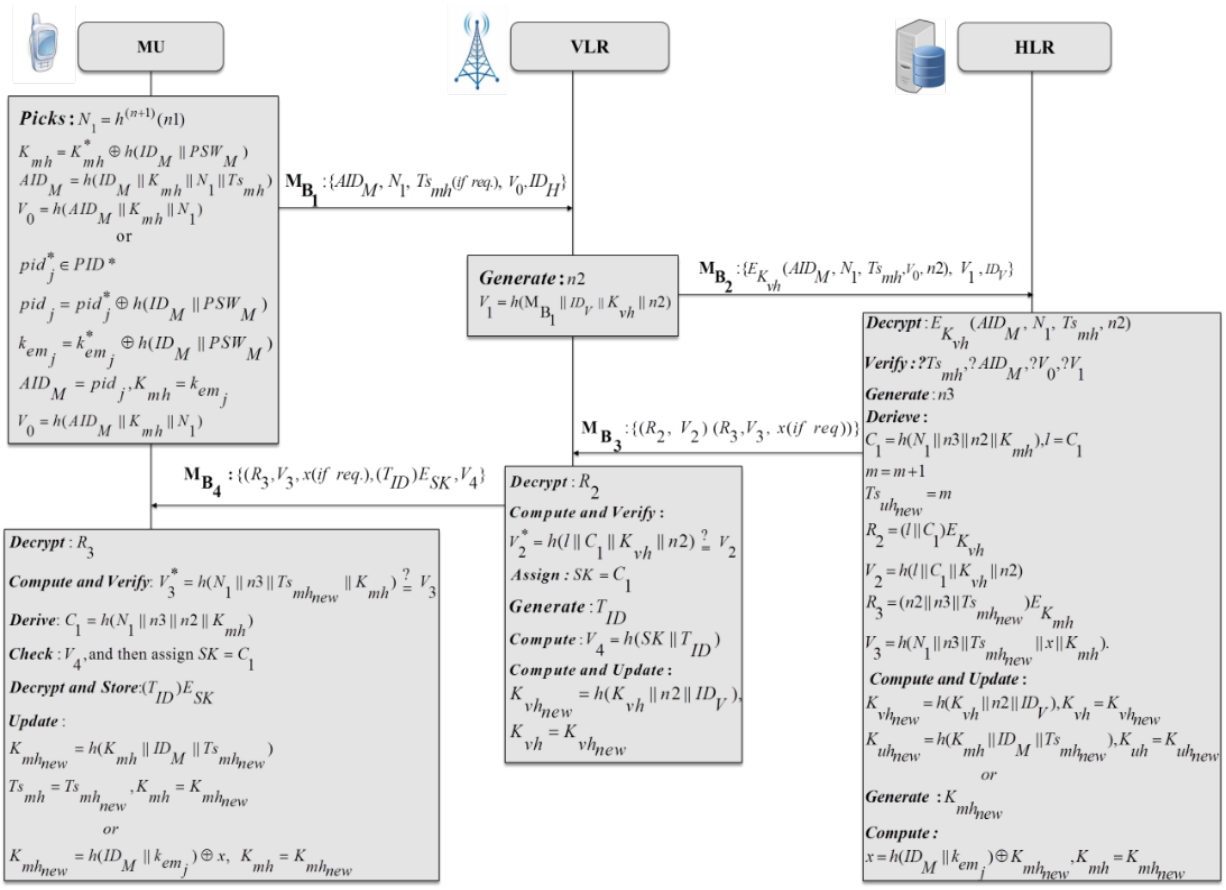

Fig. 1. Proposed on-line authentication process

$k_{em_j} = k^*_{em_j} \oplus h(ID_M \| PSW_M)$. Subsequently, the smartcard sets $AID_M \leftarrow pid_j$ and $k_{em_j} \leftarrow K_{mh}$. In this case, the MU needs not to send Ts_{mh} in M_{B_1}.

Step 2 M_{B_2} : VLR → HLR : $\{E_{K_{mh}}(AID_M, N_1, Ts_{mh}, V_0\ n2), V_1, ID_V\}$.

After receiving $\{AID_M, N_1, Ts_{mh}(if\ req.), V_0, ID_H\}$, the VLR selects a random number $n2$ and then encrypts AID_M, N_1, Ts_{mh}, and $n2$ by using its own key, i.e., $E_{K_{mh}}(AID_M, N_1, Ts_{mh}, V_0, n2)$. Hereafter, the MU computes $V_1 = h(M_{B_1} \| ID_V \| K_{vh} \| n2)$ and then sends authentication message M_{B_2} to the corresponding HLR, where the MU registered earlier, to verify the legitimacy of the MU.

Step 3 M_{B_3} :HLR → VLR: $\{(R_2, V_2)(R_3, V_3, x(if\ req))\}$.

Upon receiving M_{B_2}, the HLR decrypts $E_{K_{mh}}(AID_M, N_1, Ts_{mh}, V_0, n2)$ and at first checks whether Ts_{mh} is valid. If so, then the HLR subsequently computes and verifies whether AID_M, V_0, and V_1 are valid. If the verification is successful then the HLR generates a nonce $n3$ then derives the following, $C_1 = h(N_1 \| n3 \| n2 \| K_{mh})$, $l \leftarrow C_1$, $m \leftarrow m+1$, $Ts_{mh_{new}} \leftarrow m$, $R_2 = (l \| C_1)E_{K_{vh}}$, $V_2 = h(l \| C_1 \| K_{vh} \| n2)$, $R_3 = (n2 \| n3 \| Ts_{mh_{new}})E_{K_{mh}}$, $V_3 = h(N_1 \| n3 \| Ts_{mh_{new}} \| K_{mh})$. After that, the HLR compose a response message M_{B_3} and sends

the message to the VLR. Finally, the HLR computes $K_{mh_{new}} = h(K_{mh} \| ID_M \| Ts_{mh_{new}})$, $K_{vh_{new}} = h(K_{vh} \| n2 \| ID_V)$ and updates its database with $K_{mh_{new}}$, $K_{vh_{new}}$ and $Ts_{mh_{new}}$. In case if the Ts_{mh} is missing in the message $\mathrm{M}_{\mathrm{B}_2}$, then the HLR will first validate AID_M and then try to recognize the pid_j in AID_M. If the reorganization of the pid_j is unsuccessful, then the HLR aborts the execution of the proposed protocol and requests the MU to retry with a valid and unused pair of (pid_j, k_{em_j}). However, if the reorganization of the pid_j is successful, then the HLR selects a random a new shared key $K_{mh_{new}}$ randomly and encodes it as $x = K_{mh_{new}} \oplus h(ID_M \| k_{em_j})$ and sends x with other parameters in $\mathrm{M}_{\mathrm{B}_3}$. In this case, the V_3 is calculated as $V_3 = h(N_1 \| n3 \| Ts_{mh_{new}} \| x \| K_{mh})$.

Step 4 $\mathbf{M_{B_4}}$ **:** $\mathrm{VLR} \rightarrow \mathrm{MU}$ **:** $\{(R_3, V_3, x(if\ req.), (T_{ID})E_{SK}, V_4\}$.

On receiving $\mathrm{M}_{\mathrm{B}_3}$ from the HLR, the VLR decrypts R_2 and obtains l and C_1. Hereafter the VLR checks that weather V_2 is equal to $h(l \| C_1 \| K_{vh} \| n2)$ or not. If so, then it sets $SK = C_1$ and generates a temporary identity T_{ID} for the MU and then computes $V_4 = h(SK \| T_{ID})$. Subsequently, the VLR sends the response message $\mathrm{M}_{\mathrm{B}_4}$ to the MU. If the VLR cannot validate the response parameter, V_2 then it will terminate the execution of the protocol. Finally, the VLR updates K_{vh} with $K_{vh_{new}} = h(K_{vh} \| n2 \| ID_V)$. The loss of synchronization between the VLR and the HLR may happen and it can be detected if the VLR does not receive $\mathrm{M}_{\mathrm{B}_3}$ within a specific time-period. In that case, the VLR asks the HLR for the new secret shared key, i.e., $K_{vh_{new}}$, which will be sent to the VLR securely.

On accepting $\mathrm{M}_{\mathrm{B}_4}$, the MU decrypts R_3 and gets $n2$, $n3$, and the new transaction sequence number $Ts_{mh_{new}}$. Then, the MU calculates $h(N_1 \| n3 \| Ts_{mh_{new}} \| x \| K_{mh})$ and verifies whether it is equal to V_3. If so, then the MU derives $C_1 = h(N_1 \| n3 \| n2 \| K_{mh})$ and validates V_4. If the verification is successful, then the MU takes C_1 as the session key. Otherwise, it terminates the execution of the protocol. Finally, the MU updates his/her database with shared secret key $K_{mh_{new}} = h(K_{mh} \| ID_M \| Ts_{mh_{new}})$ and new transaction sequence number $Ts_{mh} \leftarrow Ts_{mh_{new}}$. Now, if an unused and valid pair of (pid_j, k_{em_j}) is delivered with the message $\mathrm{M}_{\mathrm{B}_4}$, the MU will also receive a new shared key $K_{mh_{new}}$ as $x = K_{mh_{new}} \oplus h(ID_M \| k_{em_j})$. The MU then computes $K_{mh_{new}}$ as $K_{mh_{new}} = x \oplus h(ID_M \| k_{em_j})$ and stores it for future communication. If any verification described above fails then this phase will be terminated. If all the verifications are successful, then we can claim that the mutual authentication between the MU and the VLR is correctly performed and a secure session key *SK* is established between them. We further depicted this phase in Fig. 2.

C. Phase III: Off-line Authentication Phase

This phase is almost similar to the protocol proposed in [6] except that in our protocol only the VLR maintains the temporary identity T_{ID} for the MU. Therefore, unlike the protocol in [6], the VLR in our protocol does not send any temporary identity T_{ID} to the HLR. This phase can be described as follows.

Step 1 $\mathrm{MU} \rightarrow \mathrm{VLR}$: $[h^{(n-i+1)}(n1) \oplus T_{ID}]E_{C_i}$

The MU selects $h^{(n-i+1)}(n1)$ and T_{ID} from its own database and delivers $[h^{(n-i+1)}(n1) \oplus T_{ID}]E_{C_i}$ to the VLR, where C_i is the session key between the MU and the VLR. The VLR computes $h^{(n-i+1)}(n1)$ using C_i and T_{ID}. Subsequently, the VLR verifies whether $h^{(n-i+1)}(n1)$ is equal to l. If $h^{(n-i+1)}(n1) = l$, the VLR updates as $l \leftarrow h^{(n-i+1)}(n1)$ and $i \leftarrow i+1$, where $i \leq n$. In this case, the VLR computes the session key as $C_{i+1} = h(l, C_i)$, selects a new random and temporary identity T_{IDnew} and accordingly updates the verification table. The VLR then delivers $[T_{IDnew} \oplus T_{ID}]E_{C_{i+1}}$ to the MU.

Step 2 VLR $\rightarrow$ MU : $[T_{IDnew} \oplus T_{ID}]E_{C_{i+1}}$.

Upon receiving $[T_{IDnew} \oplus T_{ID}]E_{C_{i+1}}$, the MU obtains T_{IDnew} by decrypting the $[T_{IDnew} \oplus T_{ID}]E_{C_{i+1}}$ using the session key C_{i+1} and subsequently updates the smartcard with the new temporary identity for next communication process.

4. SECURITY EVALUATION

In this section, we analyze our protocol to show that it holds different security properties and functionality requirements to offer a secure and flexible authentication environment for PCSs.

a. Mutual Authentication

In our protocol, the HLR authenticates the MU by examine AID_M, Ts_{ms}, and V_0 in the request message $\mathrm{M}_{\mathrm{B}_2}$. A legal MU only can construct a valid request message $\mathrm{M}_{\mathrm{B}_1}$ in the message $\mathrm{M}_{\mathrm{B}_2}$. In addition to that, if the synchronization between the MU and the VLR is lost due to various reasons, the HLR authenticates the MU by using the unused pseudo identity pid_j in AID_M and by checking whether the V_0 is equal to $h(AID_M \| K_{mh} \| N_1)$. Furthermore, the MU examines the authenticity of the HLR by checking whether the V_3 is equal to $h(N_1 \| n3 \| Ts_{mh_{new}} \| K_{mh})$. Now, the HLR authenticates the VLR by verifying V_1. similarly, the VLR authenticates the HLR by using V_1., which must be equal to $h(l \| C_1 \| K_{vh} \| n2)$. From the above discussion, it can be argued that our authentication protocol for PCSs achieves the mutual authentication attribute.

b. PAE with User Anonymity and Untraceability

Based on the literature, user anonymity falls into two categories: weak anonymity and strong anonymity. In case of weak one, an adversary has no capability to know the real identities of the mobile users, but using their temporary identities can be traced by the adversary. On the other hand, in case of the strong anonymity, the adversary should not be able to trace an entity by using their temporary identities used in different sessions. The existing standard 3G UMTS [3] cannot ensure anonymity during authentication since when current VLR refuses to accept the Temporary Mobile Subscriber Identity (TMSI) of MU, it forces him/her to provide the International Mobile Subscriber Identity (IMSI) which reveals the real identity of the MU to the VLR. In our proposed delegation-based protocol, we unitize the concept of one-time-alias identity AID_M, which will be different in each transaction. No one except the HLR knows the mobile user's real identity ID_M. Furthermore, even if the MU visits a VLR multiple times, the VLR is still unable to identify the mobile user. In this way, we ensure user anonymity along with the PAE [13] support in our proposed scheme.

c. Perfect Forward Secrecy (PFS)

In our proposed scheme, we accomplish PFS by regular updating the shared secrets K_{mh} and K_{vh}. At the end of each transaction, these keys are updated using the hash function. If any of the secret keys is compromised, because of the one-way characteristic of the hash function, adversary cannot acquire K_{mh} from $K_{mh_{new}}$ and K_{vh} from the $K_{vh_{new}}$. Accordingly, our protocol achieves the PFS of the session key [14-15] and ultimately provides the guarantee of the secrecy of the old session keys.

d. Resistance to DoS Attack

Any loss of synchronization between the MU and the HLR may bring in DoS attacks [10-12]. Usually, one of the ways to comprehend if there is any DoS attack or not, is by checking that whether the response message M_{B_4} has been received in a maximum roundtrip time. For dealing with this attack, we utilize the concept of un-linkable pseudo-IDs and emergency key pairs $(PID, K_{em}) = \{(pid_1, k_{em_1}), (pid_2, k_{em_2}), ..., (pid_n, k_{em_n})\}$. It may also happen that the adversary destroys the unsinkability by continuously listen on the communication media and interrupt the connections. It is a trade-off in wireless communications and this kind of failure can be reduced to lower limit by updating the emergency keys and the transaction sequence numbers. If all the pairs are already used, then the HLR must handover a set of pseudo-identity and emergency key pairs to the MU via a secure channel.

e. Insider Attack

Usually, it is possible that more than one user selects the same password for accessing different application servers for convenience. Now, if a privileged insider of the HLR knows the password of the MU, she/he may try to impersonate the MU by accessing other application servers where the same mobile user is a registered user. For dealing with this attack, the MU of our protocol does not submit her/his password to the HLR during registration, thus a privileged insider of the HLR could not get the password of the MU. Therefore, insider attack has been prevented by our protocol.

f. Security Assurance in Case of Lost Smartcard

In general, when a smartcard is lost or stolen by someone, then the secret parameters stored in the smartcard can be obtained through the intelligent side channel attacks [16]. In our protocol, if the smartcard of a MU is lost/stolen, the attacker does not find ID_M and PSW_M of the MU. Furthermore, the adversary cannot calculate the secrets $K_{mh} = K^*_{mh} \oplus h(ID_M \| PSW_M), AID_M = h(ID_M \| K_{mh} \| N_1 \| Ts_{mh})$ or $pid_j = pid^*_j \oplus h(ID_M \| PSW_M)$ and $k_{em_j} = k^*_{em_j} \oplus h(ID_M \| PSW_M)$ without knowing ID_M and PSW_M. Therefore, the smartcard lost attack is resisted in our protocol.

5. COMPARATIVE ANALYSIS

The main intension to design a new protocol is to eliminate the security and privacy threats found in the existing authentication protocols for PCSs and to assure lower communication and computation costs. Therefore, we compared our protocol with several contemporary similar protocols in [5,6,8,9] to provide a clear view about the merits of our protocol. To reflect the security and functionality benefits, we compared our delegation-based authentication protocols for PCSs with the related protocols in [5,6,8,9], and the comparative result is addressed in Table II. The Table II demonstrates that the required functionality and security attributes are integrated with the proposed protocol, whereas other existing protocols are lacking some of these properties. In contrast, even though all the delegation-based authentication protocols in [5,6,8,9] can ensure "mutual authentication" property, they fail to achieve some other essential functionality and security attributes including, "user anonymity", "robustness against insider attacks", "robustness against side-channel attacks", and "perfect forward secrecy" "resistance of DoS attacks", which are deeply important in order to ensure secure roaming services for PCSs.

Table II

PERFORMANCE BENCHMARKING BASED ON THE SECURITY PROPERTIES

Scheme	SP1	SP2	SP3	SP4	SP5	SP6
W.B. Lee and Yeh [5]	Yes	No	No	No	No	No
T. F. Lee et al. [6]	Yes	No	No	No	No	No
R. C. Wang et al. [8]	Yes	Yes	No	No	No	No
C.C. Lee et al. [9]	Yes	Yes	No	No	No	No
Ours	Yes	Yes	Yes	Yes	Yes	Yes

SP: Security Property; **SP1:** Mutual Authentication; **SP2:** Forward Secrecy; **SP3:** Strong Anonymity; **SP4:** Robust Against Insider Attacks; **SR5:** Robust Against Side-Channel Attacks; **SR6:** Perfect Forward Secrecy with the Resistance of DoS Attacks;

TABLE III

PERFORMANCE BENCHMARKING BASED ON COMPUTATION COST DURING THE EXECUTION OF ON-LINE AUTHENTICATION PROCESS

Scheme	Mobile	VLR	HLR
W.B. Lee and Yeh [5]	$t_{Exp1}+2*t_{Exp2}+t_{Sym}+2*t_{Hash}$	$t_{Exp1}+2*t_{Exp2}+2*t_{Sym}+t_{Hash}$	$3*t_{Sym}+3*t_{Hash}$
T. F. Lee et al. [6]	$t_{Exp1}+2*t_{Exp2}+t_{Sym}+2*t_{Hash}$	$t_{Exp1}+2*t_{Exp2}+2*t_{Sym}+t_{Hash}$	$3*t_{Sym}+t_{Hash}$
R.C. Wang et al. [8]	$t_{Exp1}+2*t_{Exp2}+t_{Sym}+3*t_{Hash}$	$t_{Exp1}+2*t_{Exp2}+2*t_{Sym}+t_{Hash}$	$3*t_{Sym}+3*t_{Hash}$
C.C. Lee et al. [9]	$t_{Exp1}+2*t_{Exp2}+t_{Sym}+3*t_{Hash}$	$4*t_{Exp1}+2*t_{Sym}+t_{Hash}$	$3*t_{Sym}+t_{Hash}$
Ours	$t_{Sym}+7*t_{Hash}$	$2*t_{Sym}+4*t_{Hash}$	$3*t_{Sym}+7*t_{Hash}$

t_{Sym} : Execution time of a symmetric key operation; t_{Hash} : Execution time of a one-way hash function; t_{Exp1} : Execution time of a modular exponential operation; t_{Exp2} : Execution time of a modular multiplication operation;

TABLE IV

COMPUTATION OVERHEAD OF THE VARIOUS CRYPTOGRAPHIC OPERATIONS

Cryptographic Operation	CPU Cycles	Execution Time
Hash operation (SHA-256)	5.63×10^{2} cpo	7.81×10^{-4} msec
Symmetric key encryption/decryption (AES-CBC)	7.56×10^{2} cpo	10.5×10^{-4} msec
Modular exponential operation	9.52×10^{6} cpo	13.22 msec
Modular multiplication Operation)		12.06 msec

$t_{Hash} \approx 5.63\times10^{2}$ (Cycle per operation); $t_{Sym} \approx 7.56\times10^{2}$ (Cycle per operation); $t_{Exp1} \approx 9.52\times10^{6}$ (Cycle per operation); $t_{Exp2} \approx 8.69\times10^{6}$ (Cycle per operation);

Here, we argued that the performance against the computation of our protocol is more efficient than previous protocols proposed in [5,6,8,9], and the comparative result is shown in Table III. In particular, our protocol does not utilize the modular exponential operation or modular multiplication, which demand high computation cost than other cryptographic operations. As an alternative, our protocol utilizes the symmetric key tools such as, hash function and encryption, which require less computation cost compared to any modular operation. To elaborate the performance of our delegation-based authentication protocol, here we have executed the different cryptographic operations, which are integrated with the design of our protocol and the protocols in [5,6,8,9] as well using Java Cryptographic Extension (JCE) [17] on a smartphone of HTC Desire as a testbed. The smartphone runs Android 2.2 mobile OS and equipped with 0.72 GHZ Arm Cortex-A8 CPU and the outcomes are summarized in Table IV. The Table IV clearly demonstrates that hash function and symmetric key encryption/decryption cause significantly less communicational overhead as compared to any modular operations. Moreover, it should be noted that, all the state of the art protocols in [5,6,8,9] exchanged six messages during the execution of the on-line phase. On the contrary, our protocol exchanged only four messages between MU, VLR, and HLR during execution of the on-line authentication phase. As a result, communication cost of the proposed protocol is much less than the protocols in [5,6,8,9]. Accordingly, we can argue that our protocol has significantly outperformed the protocols presented in [5,6,8,9] in all aspects. Therefore, our protocol is much suitable to provide secure and flexible roaming facilities to the mobile users in PCSs.

6. CONCLUSION

According to literature reviews, existing delegation-based protocols for PCS environments fail to provide some important security and functionality properties such as mutual authentication, PFS, resistance to DoS attacks, resistance to insider attack, and side-channel-attacks. Besides, the existing protocols also violate the PAE with user anonymity and untraceability, where the MU is still traceable by the old VLR, even if the MU moves to a new VLR. As a result, these protocols infringe the privacy issues of the mobile users. We put forward a new delegation based authentication protocol for PCS environments to avoid the above-mentioned issues of the mobile users. The comparative discussion made in the previous section exhibits resilience against known security threats and our protocol can even guarantee reasonable computational and communication overhead. Hence, it can be much suitable for portable communication systems.

REFERENCES

[1] C. C. Lee, I. E. Liao, and M. S. Hwang, "An efficient authentication protocol for mobile communication," *Telecommunication Systems*, vol. 46, No. 1, pp. 31-43, 2011.

[2] C. C. Lee, M. S. Hwang, and W. P. Yang, "Extension of authentication protocol for GSM," *IEE Proceedings-Communications*, vol. 150, no. 2, pp. 91-95, 2003.

[3] T. Hwang and P. Gope, "Provably secure mutual authentication and key exchange scheme for expeditious mobile communication through synchronously one-time Secrets," *Wireless Personal Communications*, vol. 77(1), pp. 197-224, 2014.

[4] P. Gope and T. Hwang, "Enhanced secure mutual authentication, and key agreement scheme preserving user anonymity in global mobile networks," *Wireless Personal Communications*, DOI: 10.1007/s11277-015-2344-z, 2015.

[5] 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.

[6] 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. http://dx.doi.org/10.1109/TWC.2009.070032.

[7] T.-Y Youn, J. Lim, "Improved delegation-based authentication protocol for secure roaming service with unsinkability," *IEEE Communication Letters*, Vol. 14, No.9, pp. 791-793, 2011.

[8] R.-C Wang, W.-S Juang, C.L. Lei, "A privacy and delegation-enhanced user authentication protocol for portable communication systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol.6, No. 3, pp. 183-190. 2011.

[9] C. C. Lee, R. X. Chang, T. Y. Chen, L. A. Chen, "An improved delegation-based authentication protocol for PCSs," *Information Technology and Control*, 2012, vol.41, No.3, pp. 258-267.

[10] P. Gope, T. Hwang, "Security Weaknesses on a Delegation-Based Authentication Protocol for PCSs," *Information Technology and Control*, Vol. 44 (3), pp. 329-333, 2015.

[11] C. H. Wang and C. Y. Lin, "An efficient delegation-based roaming payment, protocol against denial of service attacks," in Proc. 2011 *International Conference on Electronics, Communications and Control*, pp. 4136–4140, 2011.

[12] J. L. Tsai, N. W. Lo, and T. C. Wu, "Secure delegation-based authentication protocol for wireless roaming service," *IEEE Communications Letters*, vol. 16(7), 2012.

[13] A. Herzberg, H. Krawczyk, and G. Tsudik, "On travelling incognito," *IEEE Workshop on Mobile systems and Applications*, pp. 205-211, 1994.

[14] W. Diffie, P. C. van Oorshot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges," Designs, Codes and Cryptography, V.2, Kluwer Academic Publishers, 1992, pp. 107-125.

[15] P. Gope and T. Hwang, "Lightweight and Energy Efficient Mutual Authentication and Key Agreement Scheme with User Anonymity for Secure Communication in Global Mobility Networks," *IEEE Systems Journal*, Vol. 10(4), pp. 1370-1379, 2016

[16] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis", in Proc. *CRYPTO'99*, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.

[17] Oracle Technology Network, "Java Cryptography Architecture", (JCA), http://docs.oracle.com/javase/6/docs/technotes/guides/crypto/CrypoSpec.html.