Privacy-Preserving Universal Authentication Protocol for Wireless Communications

Daojing He, *Student Member, IEEE*, Jiajun Bu, *Member, IEEE*, Sammy Chan, *Member, IEEE*, Chun Chen, *Member, IEEE*, and Mingjian Yin, *Student Member, IEEE*

***Abstract*—Seamless roaming over wireless networks is highly desirable to mobile users, and security such as authentication of mobile users is challenging. In this paper, we propose a privacy-preserving universal authentication protocol, called *Priauth*, which provides strong user anonymity against both eavesdroppers and foreign servers, session key establishment, and achieves efficiency. Most importantly, Priauth provides an efficient approach to tackle the problem of user revocation while supporting strong user untraceability.**

***Index Terms*—Authentication, privacy, revocation, key establishment, wireless communications.**

I. Introduction

Mobile handled devices (e.g., notebook computer, PDA and smart phone) in wireless networks are gradually changing the way we live our life. For allowing people to get connected seamlessly using their devices without being limited by the geographical coverage of their own home networks, a roaming service should be deployed. A typical roaming scenario involves three parties: a roaming user U, a visiting foreign server V and a home server H of which U is a subscriber. When U is in a foreign network administered by V, roaming service enables U to access its subscribed services through V. There is a direct communication link between U and V and another between V and H. However, there is no direct communication link between U and H. To prevent fraudulent use of services, user authentication is a mandatory requirement. In addition, user privacy has become a serious concern in roaming services as roaming protocols may expose users' identities and locations at the user authentication phase. These considerations necessitate privacy-preserving user authentication.

A privacy-preserving user authentication scheme should satisfy the following requirements [1]: (1) **Server Authentication**: a user is sure about the identity of the foreign server. (2) **Subscription Validation**: a foreign server is sure about the identity of a user's home server. (3) **Provision of user revocation mechanism**: due to some reasons (e.g., the subscription period of a user has expired or a user's secret key has been compromised), user authentication should allow a foreign server to find out whether a roaming user is revoked. (4) **Key establishment**: the user and the foreign server establish a random session key which is known only to them and is derived from contributions of both of them. In particular, the home server should not know the session key (e.g., [2], [3]). (5) **User anonymity**: besides the user and its home server, no one including the foreign server can tell the identity of the user; and (6) **User untraceablility**: besides the user and its home server, no one including the foreign server is able to link any past or future protocol runs of the same user.

When user revocation is supported in an authentication protocol, it is more challenging to achieve user untraceability because on one hand, information is given to foreign servers to identify revoked users, but on the other hand, the information should not enable foreign servers to link other protocol runs of the revoked user. More specifically, the protocol runs involved by a revoked user before his revocation should remain anonymous and unlinkable. This is referred to as backward unlinkability in roaming service. In addition, for a time-limited revocation due to, for example, suspension of service for a period of time, the anonymity and the unlinkability of the revoked user's protocol runs after the revocation period should also be maintained. We refer to this property as forward unlinkability in roaming service. Requirement (6) includes backward and forward unlinkabilities which, until now, are unsolved problems.

In this paper, we assume that the attacker has total control over all communication channels among the user, foreign server and home server. That is, the attacker may intercept, insert, delete, or modify any message in the channels. Particularly, we consider four major types of threats to user authentication, namely, message en route threat, false mobile user threat, DoS attack and deposit-case attack [4]. The message en route threat includes that an attacker relays and/or redirects messages. The false mobile user threat includes the case where an attacker could impersonate a foreign/home server, as well as the case where mobile users under the control of an attacker collude. DoS attack refers to the overwhelming service requests from attackers in the purpose of blocking services from genuine mobile users. In deposit-case attack, the user is honest while there is a malicious server M, who will make the foreign server V to believe that the home server of the user is M without being detected by the user nor its home server.

This paper makes two main contributions: (1) We show some security weaknesses of current user authentication protocols in wireless communications. (2) We propose a privacy-

Manuscript received June 10, 2010; revised September 2, 2010 and November 9, 2010; accepted November 10, 2010. The associate editor coordinating the review of this paper and approving it for publication was D. Tarchi.

D. He, J. Bu, C. Chen, and M. Yin are with the College of Computer Science, Zhejiang University, P. R. China (e-mail: hedaojinghit@gmail.com).

S. Chan is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, P. R. China.

Digital Object Identifier 10.1109/TWC.2010.120610.101018

preserving universal authentication protocol called *Priauth*. By introducing Verifier-Local Revocation Group Signature with Backward Unlinkability (VLR-GS-BU), it can satisfy all requirements described above. Also, Priauth only requires the roaming user and the foreign server to be involved in each protocol run, and the home server can be off-line. Additionally, Priauth belongs to the class of *Universal Authentication Protocols* [2] in which same protocol and signaling flows are used regardless of the domain (home or foreign) a roaming user is visiting. This helps reducing the system complexity in practice. Furthermore, Priauth supports verifier-local revocation, which means that verifiers (i.e., foreign servers) can, based on the revocation list (*RL*) sent from the home server, check locally whether a roaming user is revoked. Note that VLR-GS-BU is not originally designed for authentication purpose and a direct application of it imposes two problems in Priauth. Firstly, it does not allow Priauth to support new group member joining after system setup. Secondly, it does not provide Priauth the single registration property commonly available in most existing authentication protocols, which requires a user only to register once at the home network before being able to access the global network. We will provide solutions to these two problems to make Priauth practical.

The remainder of this paper is organized as follows. In the next section, we first survey and analyze the related work, and then discuss their security weaknesses. Section III describes Priauth in detail. The theoretical analysis of the security properties of Priauth is provided in Section IV. Then in Section V, we discuss some important issues about our scheme and further improve it. Experimental results and performance analysis of Priauth are given in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Due to the importance of roaming service, many efficient authentication protocols have been proposed (e.g., [1]-[3], [5]-[10]). Conventionally, performing user authentication is to let the foreign server V contact the home server H who acts as a guarantor for vouching that a roaming user U is a legitimate subscriber of it. Most existing roaming protocols (e.g., [1], [3], [5]-[10]) employ this method. Unfortunately, since this method requires a foreign server to unconditionally forward any login request, valid or invalid, to the home server, attackers can easily launch DoS attacks on a home server through a foreign server. Also, these protocols cannot satisfy requirement (3) and some of them (e.g., [1], [5]-[10]) cannot achieve requirement (4). A universal authentication protocol with strong user anonymity is proposed in [2]. It only requires the roaming user and the foreign server to be involved in each protocol run, the DoS attack on home servers is thus not applicable. However, in this protocol, V uses a challenge-response approach to establish a session key with U before it authenticates U. An attacker can easily send a large volume of forged login requests to exhaust the storage and processing resources of foreign servers. Compared with other authentication methods, this protocol can provide a practical user revocation mechanism. However, contrary to their claims, we observe that the protocol fails to provide user untraceablility because once a particular user exists in the *RL* sent to V, V is able to identify all (including past and future) protocol runs which the user has and will be involved. The detailed analysis is as follows. At the beginning of a particular day, V downloads the latest revocation list *RL* which contains the trace keys of the users revoked by H. With the trace keys, V can identify whether U has been revoked. In this protocol, every user's trace key remains unchanged. Thus, once a user exists in a particular day's *RL* (i.e., once V gets a user's trace key), all protocol runs of the user are linkable to the trace key. In general, the RL is large and updated very frequently, which means that V can obtain many users' trace keys. Therefore, this weakness is serious. Obviously, their approach cannot satisfy requirement (6). According to the above analysis, all existing authentication protocols fail to meet the security requirements that a privacy-preserving authentication should satisfy.

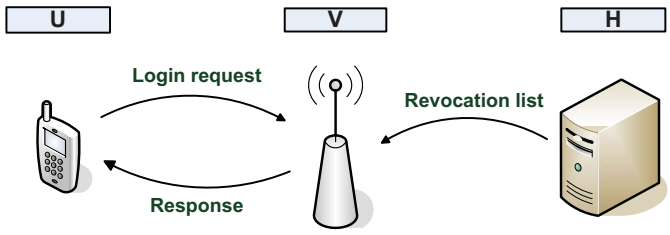

Fig. 1. The system overview of Priauth.

III. Priauth

A. Overview

Figure 1 shows the system overview of Priauth. As mentioned in Section I, it involves three kinds of participants, a roaming user U, a visiting foreign server V and a home server H. The user U who wants to access the global network firstly registers to H. When U roams into a foreign network administrated by V, U sends a login request to V. After V makes sure that U is a subscriber of H, it gives a response to U and establishes a session key with U. H periodically publishes a *RL* to foreign servers including V so that V can look up the *RL* to find out if a roaming user is revoked or not without actually knowing who the roaming user is, and the whole process should be done without any realtime involvement of H. Here we assume that the special case, in which the revocation list on a foreign server is expired and the foreign server cannot link to the home server, does not exist.

To ensure that V can identify whether U is a subscriber of H without actually knowing the identity of U and involving H, a straightforward method is the use of basic group signature. A group signature scheme is a method for allowing a member of a group to anonymously sign a message on behalf of the group. For example, a group signature scheme could be used by a subscriber of H, allowing a verifier (i.e., V here) to check if a login request was signed by a subscriber of H, without knowing the identity of the subscriber who signed it. To further support user revocation, the simplest approach is that the group manager changes and re-distributes the group public key and secret keys of all but the revoked users. However, it incurs enormous loads to non-revoked users. There is another method, where revocation messages

are only sent to verifiers. Since the signers' processing load is lower, this approach is suitable for mobile environments where mobile users anonymously communicate with the verifiers. We refer to this as the Verifier-Local Revocation (VLR) group signature approach. However, since a basic VLR group signature (e.g.,[11]) only provides one user revocation token for each user, once the tracing trapdoor of a group member is revealed, all signatures created by that member become linkable. That is, all protocol runs involved by the member become linkable.

A more suitable approach is that U signs a login request message with VLR-GS-BU. It can provide a way to trace users' signatures in individual period. Thus it satisfies requirement (6) and can remedy the security weakness of the above two group-signature approaches.

B. Priauth

We present a universal authentication protocol based on VLR-GS-BU. VLR-GS-BU is a tuple (VLR-GS.Keygen, VLR-GS.Sign, VLR-GS.Verify) of probabilistic polynomial-time algorithms: (1) VLR-GS.Keygen (N, T): The group manager runs this algorithm. This algorithm takes as input integers N, $T \in \mathbb{N}$ indicating the number of subscribers (i.e., users) and the number of time intervals, respectively. Its output consists of a master public key mpk, a vector of N subscribers' secret keys $usk = (usk[1], \ldots, usk[N])$ and a vector of $N \times T$ revocation tokens urt=$(urt[1][1], \ldots, urt[1][T], urt[2][1], \ldots, urt[2][T], \ldots, urt[N][T])$, where $urt[i][j]$ denotes the revocation token of user U_i at time interval j. (2) VLR-GS.Sign$(mpk, usk[i], j, M)$: This algorithm takes the master public key mpk, $usk[i]$, the current time interval j and a message $M \in \{0,1\}^*$, and outputs a group signature σ. (3) VLR-GS.Verify$(mpk, j, RL_j, \sigma, M)$: It takes as input mpk, the interval j, a set of revocation tokens RL_j for interval j, a signature σ, and the message M. It outputs either "valid" or "invalid". The former output denotes that σ is a correct signature on M at interval j with respect to mpk, and the signer is not revoked at interval j. Next we review a concrete VLR-GS-BU scheme of [12]. Let G be a cyclic group of large prime order p.

VLR-GS.Keygen (N, T): The group manager randomly selects a generator $g \in G$ and $\widetilde{g} \in_R G$. Additionally, it selects $h_j \in_R G$ for all $j \in [1, T]$. Then it selects $\gamma \in_R \mathbb{Z}_p^*$ and computes $w = g^\gamma$. Subsequently, it selects $x_i \in_R \mathbb{Z}_p^*$ and computes $A_i = g^{1/(\gamma + x_i)}$ for all $i \in [1, N]$. After that, it computes $B_{ij} = h_j^{x_i}$ for all i and j. The master public key mpk is $(g, \widetilde{g}, h_1, \ldots, h_T, w)$. Each subscriber's secret key $usk[i]$ is (A_i, x_i). The revocation token at interval j of subscriber with secret key (A_i, x_i) is $urt[i][j] = B_{ij}$.

VLR-GS.Sign$(mpk, usk[i], j, M)$: We assume that a signed message $M \in \{0,1\}^*$ includes the time interval j in order to bind the signature to the interval. The algorithm is as follows. (1) Select random number $\alpha, \beta, \delta \in_R \mathbb{Z}_p^*$. (2) Compute $T_1 = A_i \widetilde{g}^\alpha, T_2 = g^\alpha \widetilde{g}^\beta, T_3 = e(g^{x_i}, h_j)^\delta$, and $T_4 = g^\delta$. (3) Compute $V = SPK\{(\alpha, \beta, \delta, x_i, A_i) : T_1 = A_i \widetilde{g}^\alpha \wedge T_2 = g^\alpha \widetilde{g}^\beta \wedge T_3 = e(g^{x_i}, h_j)^\delta \wedge T_4 = g^\delta \wedge e(A_i, w g^{x_i}) = e(g, g)\}(M)$. For simplicity, the detailed description of the signature from zero-knowledge proofs of knowledge (*SPK*) is omitted in this paper. The reader can refer to [12]. (4) Output the group signature $\sigma = (T_1, T_2, T_3, T_4, V)$.

VLR-GS.Verify$(mpk, j, RL_j, \sigma, M)$: The inputs are $mpk = (g, \widetilde{g}, h_1, \ldots, h_T, w)$, the current time interval j, the revocation list RL_j that consists of $urt[i][j]$ for all revoked U_i at interval j, a target signature $\sigma = (T_1, T_2, T_3, T_4, V)$, and the message $M \in \{0,1\}^*$. This algorithm can perform two functions: (1) **Signature check**. Check that σ is valid, by checking the SPK V. (2) **Revocation check**. Check that the signer is not revoked at interval j, by checking $T_3 \neq e(T_4, B_{ij})$ for all $B_{ij} \in RL_j$.

We consider that there are multiple servers, each server manages a group of subscribers, and each subscriber could be a roaming user. Below is the system setup. (1) Each server is the group manager of an independent VLR-GS-BU scheme and has a master public key mpk generated using VLR-GS.Keygen. The master public key mpk of each server is publicly known to all other servers. In practice, this could be realized by the conventional Public Key Infrastructure (PKI). More exactly, there exists a trusted Certificate Authority (CA) who issues a digital certificate to each server, so that the certificate binds the server's identity and its master public key. For each subscriber of a server H, say U_i, U_i secretly obtains a user secret key $usk[i]$ from H during the registration phase while the vector of $N \times T$ revocation tokens is kept by H. H is called the home server of the subscriber U_i. Each server also has a signing/verification key pair (sk, pk) of a conventional digital signature method, e.g., ECDSA [13]; (2) To make efficient revocation checking, we make a small extension to the VLR-GS-BU scheme as follows. As the group manager of an independent VLR-GS-BU system, each server can set the interval unit (e.g., hour, day, month). We assume the server H sets day as the interval unit. Thus at the beginning of each day, say j, all servers except H download the latest revocation list $RL_j = \{urt[k_1][j], \ldots, urt[k_i][j], \ldots, urt[k_l][j]\}$ from H, where $1 \leq k_i \leq N$. (3) The ID and mpk of each server are publicly known to all the users who are within the network controlled by the server. This could be realized by requiring the serving network to broadcast its digital certificate to all the users currently in the network.

In the following, we describe the details of the protocol which is carried out between a roaming user U_i (whose home server is H) and a visiting foreign server V. The protocol is illustrated in Fig. 2.

1) U_i firstly chooses a random number R_u, and a temporary identity $alias$, and generates σ_U=VLR-GS.Sign$(mpk_H, usk[i], j, H \| V \| alias \| g^{R_u} \| ts)$ and then sends $\{H, alias, g^{R_u}, ts, \sigma_U\}$ to V. Here a timestamp ts is added by U_i to counter replay attacks.

2) After receiving the message, V verifies it. If the signature is invalid, V rejects it; otherwise, V chooses a random number R_v, and computes $\sigma_V = ECDSA.Sig(sk_V, m_V)$, where $m_V = H \| V \| alias \| g^{R_u} \| g^{R_v}$. Then V sends $\{V, g^{R_v}, \sigma_V\}$ back to U. Subsequently, V computes the session key $SK = (g^{R_u})^{R_v}$ and erases R_v from its memory.

3) Upon receiving $\{V, g^{R_v}, \sigma_V\}$, U verifies σ_V by running ECDSA.Ver(pk_V, m_V, σ_V). If ECDSA.Ver returns 1, U gen-

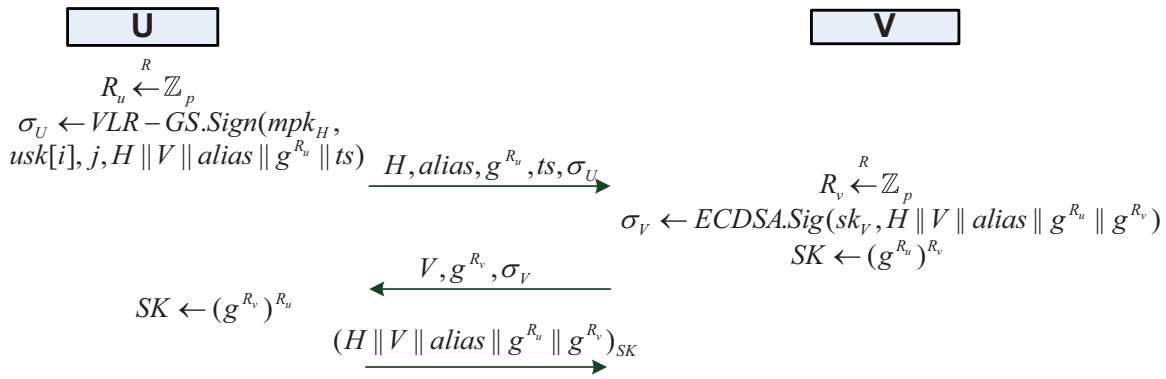

Fig. 2. The protocol run of Priauth.

erates the session key $SK = (g^{R_v})^{R_u}$ and erases R_u from its memory. After that, U generates $(H\|V\|alias\|g^{R_u}\|g^{R_v})_{SK}$ and then sends it to V. Here $(X)_K$ indicates encrypting a message X using a symmetric key K. After receiving the message, V decrypts and then verifies it. If the message is valid, V concludes that U has established a session key; otherwise, V rejects the connection.

Obviously, for $\forall j \in [1, 2, \ldots, T]$, if H hopes to revoke a particular user U_i, he simply puts the revocation token $urt[i][j]$ into RL_j. Otherwise, for $\forall j \in [1, 2, \ldots, T]$, if H allows U_i to access the global network, H does not put $urt[i][j]$ into RL_j. In addition, through simply replacing V with H, this protocol can also be used for authentication and key establishment when U is in its home network. Hence Priauth is a Universal Authentication Protocol.

IV. Security Analysis

We analyze the security of Priauth to verify whether the requirements mentioned in Section I have been satisfied. Server authentication is done by the challenge-response pair $\{\{H, alias, g^{R_u}\}, ECDSA.Sig(sk_V, \{H\|V\|alias\|g^{R_u}\|g^{R_v}\})\}$. Due to the existential unforgeability of digital signature, only V who has sk_V can generate a valid signature on U_i's freshly generated challenge $\{H, alias, g^{R_u}\}$. Since only the trusted CA can generate a valid certificate for V, and the identity of V and its verification key pk_V are included and bound by the certificate, V cannot cheat by using different verification pairs, or different IDs. Subscription validation is achieved by the message $\{\{H, alias, g^{R_u}, ts\}$,VLR-GS.Sign$(mpk_H, usk[i], j, H\|V\|alias\|g^{R_u}\|ts)\}$. Due to the existential unforgeability of the group signature, only a legitimate subscriber of H can generate a valid signature on the freshly generated sub-message $\{H, alias, g^{R_u}, ts\}$. Note that only the trusted CA can generate a valid certificate for server H, and the identity of H and its master public key mpk_H are included and bound by the certificate. Therefore, no one can cheat V. Additionally, since Priauth satisfies requirements (1) and (2), it can resist message en route and false mobile users threats.

To analyze Priauth with respect to user anonymity and untraceability. We consider two cases according to whether a roaming user U_i exists in the *RL* of H during a particular interval j. One case is user U_i does not exist in the *RL* of H. User anonymity is achieved due to the anonymity of VLR-GS-BU, which is a special group signature algorithm. V is not able to obtain the identity of the real signer since it does not have U_i's revocation token $urt[i][j]$, only U_i's home server H has. User untraceability is also achieved by the anonymity of VLR-GS-BU. The reason would become clear when readers refer to the anonymity definition for VLR-GS-BU in [12]. Here we mainly focus on the second case, where U_i exists in the *RL* of H during a particular interval j. Thus, V can obtain U_i's revocation token $urt[i][j]$ and uses it to make sure that U_i is revoked for interval j. Since the revocation token of each user evolves for every interval, V cannot link U_i's protocol run during any interval j_1 to $urt[i][j]$, where $j_1 \neq j$. That is, Priauth can preserve the anonymity and the unlinkability of U_i's protocol runs during past and future periods. According to the above analysis, Priauth can provide user anonymity and untraceability.

Priauth only requires the user and the foreign server to be involved in each protocol run, and the home server can be off-line. Thus, DoS attack on home servers is not applicable. Also, since a foreign server authenticates a user at the very beginning in the protocol execution, Priauth can mitigate DoS attack on foreign servers. For deposit-case attack, suppose a malicious server M manages to modify the user's claim and then produces a group signature to V. In this case, V will use its signing key to make signature on the identity of M and then send the signature to the user. With the verification key of V, the user can know that V does not think its home server is H. Thus, this attack can be detected by the user.

V. Discussion

A. New User Joining

New user joining is about allowing a new user to register to a server after system setup. To support dynamic participation, an authentication scheme should support new user joining. For the above protocol, however, this new user joining mechanism no longer works. A feasible new user joining mechanism is added into Priauth as follows. We assume a user U_n hopes to register to a server H during interval j_n. After verifying U_n's information, as the group manager of an independent VLR-GS-BU system, H selects $x_n \in_R Z_p^*$ and computes $A_n = g^{1/(\gamma + x_n)}$. After that, it computes $B_{nj} = h_j^{x_n}$ for all $j \in [j_n, T]$. The master public key mpk is still

TABLE I
PERFORMANCE COMPARISON BETWEEN PRIAUTH AND RELATED WORK

DoS: DoS attack resistance; BF: Provision of User Revocation with Backward and Forward Unlinkabilities

Protocols	Number of parties	Universal	Communication overhead	Single Registration	DoS	BF	User Untraceablility	Key establishment	User public key operations
HZCB [1]	3	No	2β+2δ	Yes	No	No	Yes	No	-
YHWD [2]	2	Yes	2δ	No	No	No	No	Yes	8.75ECSM +3Pairing
HCCBF [5]	3	No	2β+2δ	Yes	No	No	Yes	No	-
YWD [3]	3	No	$\geq 5\beta + 3\delta$	Yes	No	No	Yes	Yes	6.25ECSM
Priauth	2	Yes	2δ	Yes	Yes	Yes	Yes	Yes	15.75ECSM +4Pairing

TABLE II
TIMINGS FOR ECSM AND PAIRING OPERATIONS

	798MHz Processor		1GHz Processor		1.33GHz Processor		1.60GHz Processor	
	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing
Time(ms)	1.767	11.888	1.740	11.0	1.729	9.287	1.719	9.028

$(g, \widetilde{g}, h_1, \ldots, h_T, w)$. U_n's secret key $usk[n]$ is (A_n, x_n). The revocation token at interval j of user U_n is $urt[n][j] = B_{nj}$, where $j \in [j_n, T]$.

B. Home Server Update

As described in Section III.B, the lifetime of Priauth, say T_{life}, is computed as $T_{life} = T \times T_{unit}$. Here T is the number of time intervals while T_{unit} is the interval unit. Not only the length of the master public key of H but also the number of revocation tokens is linear to T. As mentioned above, the master key of H is stored on every subscriber of H while the revocation tokens are stored on H. Considering the limited storage resource of mobile devices, T should be restricted. However, to extend the lifetime of the proposed protocol, T should be large enough. Regarding this point, there exists a tradeoff. Clearly, at the end of the protocol lifetime, all users need to re-register to their home server H. In some settings, it may not be convenient for a user to re-register to his previous home server H after he leaves his home network. To support single registration as most existing authentication protocols do, we present a practical approach which removes the need of user re-registration after the protocol lifetime expires.

We assume that at interval T_1, the lifetime of a server H has expired. We also assume the number of intervals of the next lifetime of H is T_2. In addition, we assume that at interval T_1, there are N_1 subscribers, whose secret key $usk[i]$ is (A_i, x_i), respectively. Here $i \in \{1, \ldots, N_1\}$. To ensure that Priauth still runs for $\forall j \in [T_1 + 1, T_1 + T_2]$, the home server H just needs to recompute the new master public key and the revocation tokens for the N_1 subscribers. The other procedures of Priauth remain unchanged. The detailed description is as follows. As the group manager of an independent VLR-GS-BU system, H selects $h_j \in_R G$ for all $j \in [T_1 + 1, T_1 + T_2]$. Then it computes $B_{ij} = h_j^{x_i}$ for all i and j. The new master public key mpk is $(g, \widetilde{g}, h_{T_1+1}, \ldots, h_{T_1+T_2}, w)$. Each subscriber's secret key $usk[i]$ is still (A_i, x_i). The revocation token at interval j of subscriber with secret key (A_i, x_i) is $urt[i][j] = B_{ij}$. Note that g, $\widetilde{g}$ and w are unchanged. Through the conventional PKI, the new master public key of H is publicly known to all other servers. Also, the new master public key of H is distributed to its subscribers in the following way. When a subscriber U of H roams into a foreign network administrated by V after interval T_1, assuming that U somehow has not yet obtained the new master public key of H, U can obtain the new master public key by requesting H's digital certificate from V.

VI. PERFORMANCE AND IMPLEMENTATION

Table I shows the performance comparison of Priauth and related works ([1]-[5]). Note that the complexity of highly efficient operations such as hash function and symmetric encryption/dencryption operation is omitted. Here public-key operations are counted as follows: ECDSA [13] takes 1 Elliptic Curve Scalar Multiplication(ECSM) operation for signing, and 1 Multi-ECSM ($\approx$1.25 ECSM [14]) operation for verification; the Diffie-Hellman exchange takes 2 ECSM operations; and a public key encryption takes 2 ECSM operations. The ECSM operation of OpenSSL [15], an open source implementation of the publicly available SSL [16] specification, has been introduced into the implementation of Priauth. The implementation results on ECSM and Pairing [17] are summarized in Table II. we perform the same experiment ten thousand times and take an average over them. From Table I, it is easy to visualize that a successful user authentication in Priauth requires 15.75ECSM and 1 Pairing computation (plus 3 Pairing computations that can be pre-computed) on a roaming user. We assume the access device of a roaming user runs on a 798 MHz processor, thus it takes 39.7 ms (plus 35.7 ms pre-computed). Currently, the clock frequency of most Laptop PCs, PDAs and smartphones is greater than 700 MHz. Therefore, Priauth is efficient to be employed on most mobile devices. For new user joining, it just takes $(T - j_n + 2)$ ECSM computations on H while the new user does not need to do any computations. Suppose that a new user hopes to subscribe a 365-day service, this incurs 366 ECSM computations on H. Additionally, home server update takes $(1 + N_1 + N_1 \times T_2)$ ECSM computations on H. In general, a foreign server or home server is a powerful server (i.e., mainframe), hence the resource consumption on them is negligible. For communication overhead, we assume

that the expected authentication message delivery cost between the foreign server and the home server is β unit and that between the roaming user and the foreign agent is δ unit, respectively. As shown in Table I, same as the scheme of [2], Priauth outperforms all other protocols on communication overhead.

VII. Conclusion

In this paper, we have proposed a novel protocol to achieve privacy-preserving universal authentication for wireless communications. The security analysis and experimental results show that the proposed approach is feasible for real applications.

VIII. Acknowledgements

This work was supported by National Science Foundation of China (Grant No. 61070155), Program for New Century Excellent Talents in University (NCET-09-0685), a grant from the Research Grants Council of the Hong Kong SAR, China[Project No. City U 111208].

References

[1] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Commun.*, 2010, doi:10.1016/j.comcom.2010.02.031.

[2] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168-174, 2010.

[3] G. Yang, D. S. Wong, and X. Deng, "Anonymous and authenticated key exchange for roaming networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3461-3472, 2007.

[4] G. Yang, D. Wong, and X. Deng, "Deposit-case attack against secure roaming," in *Proc. ACISP'05*, 2005.

[5] D. He and S. Chan, "Design and validation of an efficient authentication scheme with anonymity for roaming service in global mobility networks," *Wireless Personal Commun.*, 2010, doi:10.1007/s11277-010-0033-5

[6] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 734-742, 2005.

[7] C. C. Lee, M. S. Hwang, and I. E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Trans. Consumer Electron.*, vol. 53, no. 5, pp. 1683-1687, 2006.

[8] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Commun. Lett.*, vol. 12, no. 10, pp. 722-723, 2008.

[9] J.-L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no. 3-4, pp. 115-121, 2008.

[10] H.-C. Hsiang and W.-K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118-1123, 2009.

[11] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proc. ACM CCS'04*, pp. 168-177, 2004.

[12] T. Nakanishi and N. Funabiki, "Verifer-local revocation group signature schemes with backward unlinkability from bilinear maps," in *Proc. ASIACRYPT'05*, LNCS, vol. 3788, pp. 533-548, 2005.

[13] ANSI X9.62 "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)," 1999.

[14] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press LLC, 1997.

[15] "OpenSSL," http://www.openssl.org.

[16] "SSL 3.0 Specification," http://wp.netscape.com/eng/ssl3.

[17] "Pairing based cryptographyłbenchmarks." [Online]. Available: http://crypto.stanford.edu/pbc/.