9-2007

Anonymous and authenticated key exchange for roaming networks

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Duncan S. WONG

Xiaotie DENG

Citation

YANG, Guomin; WONG, Duncan S.; and DENG, Xiaotie. Anonymous and authenticated key exchange for roaming networks. (2007). *IEEE Transactions on Wireless Communications*. 6, (9), 3461-3472.
Available at: https://ink.library.smu.edu.sg/sis_research/7402

Anonymous and Authenticated Key Exchange for Roaming Networks

Guomin Yang, Duncan S. Wong, *Member, IEEE*, and Xiaotie Deng, *Senior Member, IEEE*

***Abstract*— User privacy is a notable security issue in wireless communications. It concerns about user identities from being exposed and user movements and whereabouts from being tracked. The concern of user privacy is particularly signified in systems which support roaming when users are able to hop across networks administered by different operators. In this paper, we propose a novel construction approach of anonymous and authenticated key exchange protocols for a *roaming* user and a visiting server to establish a random session key in such a way that the visiting server authenticates the user's home server without knowing exactly who the user is. A network eavesdropper cannot find out the user's identity either (*user anonymity*). In addition, visited servers cannot track the roaming user's movements and whereabouts even they collude with each other (*user untraceability*). Our construction approach is generic and built upon provably secure two-party key establishment protocols. Merits of our generic protocol construction include eliminating alias synchronization between the user and the home server, supporting joint key control, and not relying on any special security assumptions on the communication channel between the visiting server and the user's home server. Our protocol can also be implemented efficiently. By piggybacking some message flows, the number of message flows between the roaming user and the visiting server is only three. As of independent interest, we describe a new practical attack called deposit-case attack and show that some previously proposed protocols are vulnerable to this attack.**

***Index Terms*— Anonymity, untraceability, privacy, authenticated key exchange, roaming.**

I. Introduction

With the advancement and tremendous development of computer networks and telecommunications systems, user mobility is rapidly becoming an important network feature nowadays, especially in wireless networks. In one scenario, a user originally subscribed to a network can travel to another network administered by a different operator and access services provided by this network as a visiting user or a guest. One advantage of this technology is that users can enjoy a much broader coverage in terms of services or geographical areas without being limited by that of their own networks. This capability is called *roaming*.

A typical roaming scenario involves three parties: *a roaming user*, *a foreign/visited server* and *a home server*. The roaming user, who is a *subscriber* of the home server, is now in a network administered by the foreign server. In cellular networks [1], [2], [3], roaming services are widely deployed. However, it is also especially evident that a number of important security issues and concerns on user privacy have been come into view. The latest generation, 3GPP[1], is also urging roaming services to be provided with a more promising assurance on the privacy of mobile users. Foremost among them are user anonymity and untraceability.

Based on the general interest of roaming users, it is desirable to keep the users anonymous from eavesdroppers as well as the foreign server (*user anonymity*) unless the identity information becomes critical, for example, in some emergency situations or special applications. Tracking a mobile unit's movements may also expose the identity of a roaming user. One single exposure of the identity of a user will lead to the exposure of all other sessions, both past and future, if all the roaming sessions corresponding to the user can be linked. Hence it is also important to make sure that no one would be able to tell if two roaming sessions are corresponding to the same mobile unit or not (*user untraceability*).

Besides cellular networks, there are many other roaming networks sharing the same desires of user privacy. In [4], Ateniese, et al. gave two examples of roaming applications which prefer upgrades on user anonymity and untraceability. One is the inter-bank ATM networks and the other is the credit card payment systems. Ideally, a user should not have to reveal anything to the serving network (i.e. the foreign server) other than the confirmation of his good standing with respect to his ATM card or credit card issued by his home server. However, current systems are having users give out their personal information inevitably. In addition to these systems, there are many other roaming networks emerging which share similar demands on user privacy. For example, hopping across meshed WLANs (Wireless Local Area Networks) administered by different individuals, joining and leaving various wireless ad hoc networks operated by different foreign operators, etc.

Besides user anonymity and untraceability, data confidentiality and authenticity are usually needed to protect communications between the user and the foreign server, which is providing services to the user, against adversaries which include eavesdroppers, other users and network servers, and even the user's home server. It may look strange of protecting data exchanged between the user and the foreign server from being viewed by the home server offhand. But it will become clear when considering that services are provided by the foreign server to the user but not to the home server. Also the home server is called in only as a guarantor for giving a promise that the user is indeed a legitimate subscriber of the

Manuscript received February 27, 2006; revised January 28, 2007; accepted June 3, 2007. The associate editor coordinating the review of this paper and approving it for publication was X. Zhang. This work was supported by grants from CityU Project Nos. 7001844, 7001959 and 7002001.

The authors are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (email: duncan@cityu.edu.hk).

Digital Object Identifier 10.1109/TWC.2007.06020042.

[1] http://www.3gpp.org

home server. For example, in the WLAN Roaming, when a user accesses the Internet through a foreign server, the user may not want his home server to know which network sites he is visiting. This is one of the key privacy issues that most current systems reviewed in Sec. II cannot solve.

Anonymous and Authenticated Key Exchange for Roaming (AAKE-R)

We propose a new notion in the area of authenticated key exchange. AAKE-R is a protocol involving three parties: a user and two servers, namely a home server and a foreign server. The home server can be online or offline. If the home server is online, we mean that the home server is involved in a protocol run and hence the protocol is a three-party one. Otherwise, it is a two-party protocol between the user and the foreign server. No matter which type of protocols is, the following five properties should be achieved simultaneously in one protocol run.

1) (Server Authentication) The user is sure about the identity of the foreign server.
2) (Subscription Validation) The foreign server is sure about the identity of the home server of the user.
3) (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 obtain the session key.
4) (User Anonymity) Besides the user and the home server, no one including the foreign server can tell the identity of the user.
5) (User Untraceability) Besides the user and the home server, no one including the foreign server is able to identify any previous protocol runs which have the same user involved.

Since this notion is very useful in the roaming scenario for users to travel from their home servers to foreign servers anonymously while at the same time establishing secure session keys with the foreign servers, we call such a scheme as an Anonymous and Authenticated Key Exchange for Roaming Networks (AAKE-R).

We will formulate and explain why the five properties above are enough for most practical applications of anonymous roaming. We will also review some related schemes and show that they do not satisfy some of these properties. Among these properties, Subscription Validation is the one which is most difficult to achieve. Many previous schemes have been found to be unable to achieve this property as they originally believed so. In particular, we show that the schemes proposed by Samfat et al. [5] and Go and Kim [6] cannot achieve Subscription Validation due to a new practical attack called *Deposit-Case Attack*.

We also propose an construction of AAKE-R scheme by using secure authenticated key exchange protocols as building blocks. Merits of our protocol include eliminating alias synchronization between the user and the home server, supporting joint key control, and not relying on any special security assumptions on the communication link between the foreign server and the home server. We will explain how our protocol can achieve all the five properties.

Organization. The rest of the paper is organized as follows. In Sec. II, some previous schemes related to anonymous roaming are reviewed. This is followed by the formalization and explanation on the security requirements of AAKE-R schemes in Sec. III. In Sec. IV, the deposit-case attack against Subscription Validation is described and its importance on the security of roaming protocols is explained. In Sec. V, an AAKE-R scheme is constructed using provably secure authenticated key exchange protocols as building blocks. The paper is concluded in Sec. VI.

II. Related Work

There had been a number of works on anonymity for mobile communications [4], [5], [7], [6]. In [4], [5], several levels of privacy requirements and corresponding protocols were proposed. The basic idea is to have an *alias* associated to the user which appears unintelligible to anyone except the home server. When the user requests connection to a foreign server, the user presents the alias and indicates a server to be his home server. The foreign server then forwards the alias to the claimed server for verification. Since the alias looks unintelligible, the foreign server cannot get any information about the user except the identity of his home server. This technique is commonly used for providing user anonymity in mobile communications [7], [6].

This technique has a major tradeoff. To provide user untraceability, the alias has to be renewed every time after being used. One of the issues incurred is about the alias synchronization between the user and the user's home server. The synchronization may be lost when the communication link is accidentally broken or when some state information of either party is corrupted.

The protocols of [4], [5] does not provide a way for the roaming user to authenticate the foreign server. In other words, their protocols do not satisfy Property 1, Server Authentication, as described above. Furthermore, the protocols of [4] has no key establishment between the roaming user and the foreign network. In other words, the protocols do not satisfy Property 3, Key Establishment.

Also note that providing anonymous roaming but no authenticated key establishment between the user and the foreign server can already be achieved using credential or pseudonym systems [8], [9] if none of the communicating parties is resource constrained. Such a system allows users to obtain credentials from their home servers and anonymously demonstrate possession of these credentials as many times as necessary without involving their home servers. It can also be achieved by using some e-cash mechanisms which hide user's identity. Hence, the protocols of [4] do not have noticeable advantage on providing user anonymity and untraceability when compared with the alternatives mentioned, other than some performance gain and computational complexity reduction.

Conventional pseudonym/credential systems [8], [9] do not support key establishment. It is not obvious to extend such systems to anonymous and authenticated key establishment protocols since Subscription Validation and Key Establishment

are generally not considered in these systems. In addition, the performance of these systems is poor when compared with authenticated key establishment protocols, which are generally required to be very efficient to implement, especially when considering the low-power wireless communication devices.

In each of the protocols of [5], there is a session key established for each connection between a user and a foreign server. However, besides the "Homeless" roaming protocol, the session key is also known to the user's home server. As already explained, this is undesirable. Besides this undesirable feature, all the protocols of [5] have the key value remains unchanged among all the sessions established. In Sec. III-B, we will also see that those protocols in [5] cannot provide Subscription Validation due to a new attack called deposit-case attack. This attack shows that an honest foreign server can be cheated to believe that a malicious server is the home server of a roaming user without being noticed by the user nor the original home server of the user.

In [7], a related scenario to roaming was discussed and three protocols were proposed for authenticated key establishment between two mobile units, each subscribed to a distinct server. Their protocols protect the identities of both mobile units from eavesdroppers, other mobile units and even the servers. However, there are two issues. First, the session key in their protocols is generated by the foreign server and then transferred to mobile units. Hence their protocols are key transport protocols [10]. Also, recent results showed that all the three protocols cannot preserve the privacy of the mobile units [11].

In [6], an authenticated key exchange protocol was proposed for anonymous roaming on wireless networks. Their protocol is targeted to protect the mobile unit's identity from all entities other than its home server and the serving foreign server (does not satisfy Property 4, User Anonymity). According to results from [11], it was found that a malicious server which is not communicating with the mobile unit can launch an active attack to reveal the mobile unit's identity. In addition, we find that their protocol cannot provide Subscription Validation as it is also vulnerable to the deposit-case attack [12].

Besides the protocols mentioned above, [13], [14] proposed some other authenticated key establishment protocols for roaming. However, they do not support user privacy.

In this paper, we propose a novel construction approach of AAKE-R. Our construction satisfies all the five properties specified in the previous section. It does not need any alias mechanism for providing anonymity (so there is no synchronization issue between the roaming user and his home server). In addition, the session key established between the roaming user and the foreign server is not known to the home server. In the following, we give more details to the five properties / security requirements for AAKE-R.

III. Security Requirements of AAKE-R

We first define some notations and system-wide parameters. Let k be a system-wide security parameter. There are two types of entities. Let $\mathcal{C}(k) = \{C_1, \cdots, C_{Q_1(k)}\}$ be the set of users (clients) in the system and $\mathcal{S}(k) = \{S_1, \cdots, S_{Q_2(k)}\}$ be the set of servers in the system, where Q_1 and Q_2 are some polynomials and $C_i, S_j \in \{0,1\}^k$ are the corresponding identities, for $1 \leq i \leq Q_1(k)$ and $1 \leq j \leq Q_2(k)$. We assume that each entity is already associated with a public key pair of some standard encryption scheme or signature scheme.

Subscription: The term 'subscribe' is commonly used for describing some special relationship between a user and a server without clear definition. In this paper, we focus on anonymous roaming scenario which requires us to formalize the meaning of subscription. Below is the intuition of subscription.

> "A user is *subscribed* to a server called the home server of the user if the server has the privilege to get access to the real identity of the user and track the user's movements and whereabouts on networks, to which the user has visited, no matter those visited networks are administered by other servers or by itself."

For simplicity, we assume that each user has subscribed to one and only one server, and the subscription is persistent. Hence scenarios related to changing subscriptions of users are excluded. For example, suppose a user changes his subscription from one server to another but using the same secret in conducting authenticated key establishment with the servers. Then the original network may be able to identify the user when he subscribes to a new server. This is not considered in our system.

Definition 1 (Subscribe): Given a security parameter k, 'subscribe' is a computable function f from $\mathcal{C}(k)$ into $\mathcal{S}(k)$. We say that C_a is 'subscribed' to S_h if $f(C_a) = S_h$ where $C_a \in \mathcal{C}(k)$ and $S_h \in \mathcal{S}(k)$.

By using the terminologies of mobile communications, S_h is said to be the home server of C_a and S_i is a foreign server of C_a for all $i \neq h$. We also assume that the inverse f^{-1} is computable. Hence for any $S_h \in \mathcal{S}(k)$, $f^{-1}(S_h)$ is the set of all $C_a \in \mathcal{C}(k)$ such that $f(C_a) = S_h$. In the following, we implicitly pass in the subscribe function f, server space $\mathcal{S}(k)$ and user space $\mathcal{C}(k)$ to each algorithm defined.

We now begin to define and describe the five properties / security requirements of an AAKE-R protocol. These five properties which are to be given in the following subsections are: *Server Authentication*, *Subscription Validation*, *Key Establishment*, *User Anonymity and User Untraceability*.

A. Server Authentication

For a user C_a, server authentication is to allow C_a to make sure that the communicating foreign server S_v is the one C_a is intended to connect to. This property provides an assurance to a roaming user on the identity of the visiting foreign server, and is very important in practice. In contrast to some previously proposed schemes [4], [5] which do not support server authentication, we stress that the security requirement is necessary. For example, different foreign servers may charge differently for the services they provide. The user would like to choose and be sure that he is obtaining the services provided by a specific foreign server. Another example, the user may trust some of the foreign servers but not the others. The user does not want to establish a connection with an untrusted foreign server and leak personal information to that server.

This requirement becomes increasingly important in wireless networks as impersonation attacks are much easier to launch when compared with wired networks.

In cellular networks, some old roaming systems such as GSM (Global Systems for Mobile Communications) [1] do not support server authentication. This requirement has been noted and addressed in new systems, such as the 3GPP [3].

B. Subscription Validation

From a foreign server S_v's perspective, S_v only needs to find out the identity of the user's home server S_h. That is, given S_h, S_v is to make sure that $C_a \in f^{-1}(S_h)$ without actually recovering C_a (the identity of the user). This can be considered as a replacement of client authentication.

Since user anonymity and untraceability are needed, the identity of the user should not be exposed to the foreign server. For facilitating billing, access control or other subscription oriented applications, subscription validation is needed. For example, a server has a security policy which only allows users subscribed to one particular server to access its services but does not allow users subscribed to another particular server to do so. Another example, a foreign server may charge differently for users subscribed to different servers.

In most of the current roaming protocols, Subscription Validation is done in two steps.

1) The roaming user A claims that a particular server H is A's home server.
2) That particular server, H, is then called in as a guarantor by the visiting foreign server V for giving a promise (as a one-time unforegable credential) that A is a legitimate subscriber of H. H generates a credential only after authenticating A.

This mechanism can effectively prevent a malicious user B, who is not subscribed to a server H, from making V believe that B is a legitimate subscriber of H. However, this mechanism only provides half of the measures towards Subscription Validation. It implicitly assumes that all servers are honest, that is, H generates credentials only under the fact that $A \in f^{-1}(H)$. This assumption may not be justified in roaming scenarios. There is no reason to restrict that no server would cheat. On the contrary, in Sec. IV, we elaborate in detail on how undesirable it would be when A's claim of home server H is changed by an adversary to a malicious server, say H_e, when V receives it, and H_e cheats. In particular, we describe a new practical attack called Deposit-Case Attack and show that some previous protocols do not satisfy Subscription Validation due to this new attack.

C. Key Establishment

In most applications, it does not seem to be very useful if a protocol only provides authentication but no key establishment. This is because an authentication-only protocol only provides authenticity between the intended parties when running the protocol. It does not provide any authentication after the protocol is finished. An authenticated key establishment protocol, instead, allows two intended communicating parties to establish a secret session key which is known only to the two parties. Hence after the protocol is finished, the two parties can use the session key to communicate securely and in an authenticated way (i.e. a secure communication channel).

The user C_a and the foreign server S_v establish a random session key $\sigma \in \{0,1\}^k$ which is known only to them and is derived from contributions of both of them. In particular, S_h should not obtain σ.

> **Joint Key Control:** The value of σ should not be controlled or chosen solely by any single one of the two communicating parties. This requirement was first pointed out as *joint key control* by Mitchell et al. [15], that is, the protocol should be designed to prevent either party from choosing the key value. It was noted that one of the communicating parties can arbitrarily select up to s bits of the session key and effectively control the value on these bits by choosing about 2^s random values, if the key establishment scheme is lack of joint key control. This may introduce some concern such as one party can force the use of an old key.

Also note that the home server S_h of C_a should not obtain σ. Consider the user accessing a web using resources provided by the foreign server. The home server is only called in for billing purposes. There is no reason to allow the home server to also see the data exchanged between the user and the foreign server. As a counterexample, the protocol of Samfat, et al. reviewed in Sec. III-B does not satisfy this requirement.

D. User Anonymity and User Untraceability

User Anonymity: Besides the user C_a himself and his home server S_h, no one including the foreign server can tell the identity of the user.

In the real world, there may be many different ways for finding out a user's identity. For example, attackers can acquire some geographical information from the radio signal emitted by a user to track the user's movements. Another example is that a careless user may reveal his identity in his communication with a server (in some higher protocol levels). Also, after completing a key establishment process, there is barely any control on how the session key is going to be used. For example, the user can simply send his messages in clear and these messages may contain his identity information. Hence we focus on the user anonymity and untraceability of the AAKE-R protocol only by limiting the information available to attackers to just the transcripts of AAKE-R protocol runs.

We define user anonymity in the sense that an attacker does not gain any advantage in telling who a user is from the transcripts of all AAKE-R protocol runs. Let an attacker be a pair $(\mathcal{A}, \mathcal{F})$ of probabilistic polynomial-time (PPT) algorithms, each in the first component of its inputs. Let a transcript be a list of all corresponding messages transmitted over the network for one protocol run. Let $\mathcal{T}$ be the set of pairs of transcripts and users' identities of all AAKE-R protocol runs of all users. The total number of protocol runs for all users in the model is restricted to at most $Q_3(k)$ for some polynomial Q_3. Given $\mathcal{C}, \mathcal{S}$ and f, define

$$\begin{aligned}\mathbf{Adv}^{\mathcal{A}}_{\mathcal{C},\mathcal{S},f}(k) = |\ &\Pr[\mathcal{A}(1^k, S_h, \mathcal{T}, T_a) \rightarrow C_a| \\ &(S_h, C_1, \cdots, C_{Q_3(k)}) \leftarrow \mathcal{F}(1^k, \mathcal{S}, \mathcal{C}), S_h \in \mathcal{S}(k), \\ &C_1, \cdots, C_{Q_3(k)} \in f^{-1}(S_h), C_a \in_R f^{-1}(S_h)] \\ &- \frac{1}{|f^{-1}(S_h)|}| \end{aligned} \quad (1)$$

to be the advantage of $\mathcal{A}$ in identifying the user over wild guess, where T_a is a new transcript corresponding to user C_a; $C_1, \cdots, C_{Q_3(k)}$ correspond to the users of the transcripts in $\mathcal{T}$. We say that the AAKE-R protocol provides user anonymity against attacker $(\mathcal{A}, \mathcal{F})$ if $\mathbf{Adv}^{\mathcal{A}}_{\mathcal{C},\mathcal{S},f}(k)$ is negligible for all sufficiently large k. A function ϵ is negligible if for every constant $c \geq 0$, there exists an integer k_c such that $\epsilon(k) < k^{-c}$ for all $k \geq k_c$.

In Eq. (1), $\mathcal{F}$ arbitrarily picks a home server S_h. $\mathcal{A}$ is then given S_h, and a couple of transcripts of previous protocol runs whose corresponding users are also chosen by $\mathcal{F}$. The objective of $\mathcal{A}$ is to guess which user has involved in a new transcript denoted by T_a, which corresponds to a subscriber C_a chosen uniformly at random from $f^{-1}(S_h)$. As defined by Eq. (1), an anonymous protocol should not allow $(\mathcal{A}, \mathcal{F})$ to have a non-negligible advantage in finding out the user identity over random guessing.

User Untraceability: Besides the user C_a himself and his home server S_h, no one including the foreign server is able to identify any previous protocol runs which have the same user involved.

The definition above (Eq. 1) also includes user untraceability. Notice that all other transcripts and the corresponding users' identities are known to the attacker. If the AAKE-R protocol is traceable, the attacker can link T_a to at least another transcript in $\mathcal{T}$ that corresponds to the user and therefore, be able to identify the user. Then Eq. (1) is non-negligible.

Hiding the Home Server: In the definition above, we assume that the home server of the user is known to attackers, which also include the foreign server. Typically, when a user requests a connection to a foreign server, the server needs to verify that the user is entitled to services, and/or charge the user for these services with incurred profit shared with the user's home server. Besides, in order to obtain some basic service, such as forwarding user's incoming data to the user's home server, the user's home server also needs to be identified by the foreign server. In [5], the authors also mentioned that providing privacy of level C_4 (i.e. hiding the identity of the user's home server from the foreign server) or higher will cause undesirable problems such as handling incoming calls in most of the telecommunication systems.

On the other hand, the inclusion of S_h in Eq. (1) will become undesirable when we consider the attacker to be a non-serving foreign network or just an eavesdropper. Also, the consideration above will no longer apply. In this case, S_h should be removed from the inputs of $\mathcal{A}$, S_h should be chosen randomly from $\mathcal{S}(k)$, and the advantage should be compared against the wild guess over the entire set of users, that is, $\mathcal{C}(k)$.

IV. Deposit-case Attack

Suppose there is an honest roaming user C_a trying to connect to an honest foreign server. Suppose the home server of C_a is S_h, that is, $f(C_a) = S_h$. The deposit-case attack works in such a way that during the communication between a foreign server and C_a, the foreign server would be cheated to believe that a malicious server S_e (where $S_e \neq S_h$) is the home server of C_a without being noticed by C_a nor the original home server S_h. In other words, a malicious server manages to conjure a credential and claims to the foreign server that the roaming user is subscribed to it. If this attack works on a protocol, then obviously this protocol fails to achieve Subscription Validation.

(Different from Unknown Key Share Attack). Deposit-case attack is a special kind of man-in-the-middle attacks. It is different from the Unknown Key Share Attack [16]. An unknown key share attack only applies to key agreement protocols [10]. It makes one party A believe that a session key is shared with a party B when it is in fact shared with another party C. A roaming protocol is more than a key agreement protocol. The deposit-case attack will make the mobile user believe that the foreign server V has obtained the identity of his home server (i.e. H) when V has in fact obtained the identity of another server which is malicious.

(Practical Significance). Consider an inter-bank ATM system where a customer (a roaming user) comes to an ATM terminal[2], which is not operated by the customer's bank, and deposits some money to his bank account. The ATM terminal can be considered as a foreign server. Suppose an AAKE-R protocol is carried out among the user, the foreign server and the home server (the customer's bank) and is vulnerable to the deposit-case attack. We can see that the deposit-case attack would lead the foreign server to transfer money to a malicious server (i.e. another bank which is different from the customer's bank).

Apparently, this new attack was not known in the past. Several related schemes have recently been found vulnerable to this attack [12]. In the following, we show that two anonymous roaming protocols cannot provide Subscription Validation due to this new attack. We emphasize that the purpose of deposit-case attack is to compromise a protocol with respect to authentication rather than anonymity. Since our focus on this paper is on anonymous roaming, we illustrate the attack on two related protocols only. In [12], we can see that a roaming protocol without anonymity should also be secure against deposit-case attack.

A. The Basic Protocol of Samfat et al.

In [5], the authors proposed several authentication protocols with different levels of privacy. Besides the "Homeless" authentication protocol, in which the home server is not involved, the other two protocols use the traditional two-step Subscription Validation mechanism but apparently do not consider the scenario that servers may cheat, as mentioned in

[2]For example, an ATM terminal with Visa/PLUS or Mastercard/Cirrus sign on.

Sec. III-B. However, from the context of the paper, the authors do not assume that all servers are honest either.

The protocol uses the following two functions as building blocks: $Token$ and $TICK$. $Token$ is computed by applying a block cipher $\mathcal{E}$ (such as AES [17]) under a symmetric key K_{ab} over three inputs, say, a random number (nonce) N_a, a timestamp T_a and the identity of message originator A. The function is denoted by

$$Token_{K_{ab}}(A, T_a, N_a) = \mathcal{E}_{K_{ab}}(A \ \oplus \ \mathcal{E}_{K_{ab}}(T_a \ \oplus \ \mathcal{E}_{K_{ab}}(N_a))).$$

$TICK$ is called a ticket which is used for an initiator A to send a session key K_s to a responder B. The key is intended to be shared with a third party C. This is denoted by

$$TICK_{K_{ab}}(A, B, C, K_s) = Token_{K_{ab}}(N_a \oplus C, \ N_b, \ N_a \oplus A) \oplus K_s$$

where N_a and N_b are nonces.

The Basic Protocol of Samfat et al. [5] consists of four message flows among a roaming user C_a, a foreign server S_v and C_a's home server S_h. The fourth message flow is optional. In Table I, we summarize the notations used in the protocol description.

TABLE I
NOTATIONS

Symbols	Meaning
C_a	a roaming user
S_h	C_a's home server
S_v	a foreign server
P_h, P_v	public keys of S_h and S_v, respectively
$\mathcal{E}$	public-key encryption algorithm
N_u, N_a, N_b, N_r, N_v	nonces
K_a	long-term key shared by C_a and S_h
K_{vh}	a long-term key shared by S_v and S_h
H	one-way hash function

In the following, we review the protocol with the first three message flows. We will consider the fourth message later.

$$\begin{array}{lcl} C_a \Rightarrow S_v & : & S_h, \ alias = \mathcal{E}_{P_h}(N_u \ || \ N_u \oplus C_a), \\ & & AUTH_{av} = [N_a, T_a, Token_{K_{av}}(alias, T_a, N_a)] \\ S_v \Rightarrow S_h & : & alias, \ \mathcal{E}_{P_h}(N_r \ || \ N_r \oplus S_v), \\ & & AUTH_{vh} = [N_v, AUTH_{av}, Token_{K_{vh}}(S_v, \\ & & \qquad AUTH_{av}, N_v)] \\ S_v \Leftarrow S_h & : & \mathcal{E}_{P_v}(N_r), \ TICK_{K_{vh}}(S_h, S_v, alias, K_{av}) \end{array}$$

where $K_{av} = H(C_a||S_v||K_a)$.

Deposit-Case Attack: Consider a malicious server S_e which intercepts the message sent from C_a to S_v and modifies the message for claiming that C_a is its subscriber.

$$\begin{array}{lcl} C_a \Rightarrow S_e & : & S_h, \ alias = \mathcal{E}_{P_h}(N_u \ || \ N_u \oplus C_a), \\ & & AUTH_{av} = [N_a, T_a, Token_{K_{av}}(alias, T_a, N_a)] \\ S_e \Rightarrow S_v & : & S_e, \ alias, \ AUTH'_{av} = [N_e, T_e, \\ & & \qquad Token_{K'}(alias, T_e, N_e)] \\ S_v \Rightarrow S_e & : & alias, \ \mathcal{E}_{P_e}(N_r \ || \ N_r \oplus S_v), \\ & & AUTH_{ve} = [N_v, AUTH'_{av}, \\ & & \qquad Token_{K_{ve}}(S_v, AUTH'_{av}, N_v)] \\ S_v \Leftarrow S_e & : & \mathcal{E}_{P_v}(N_r), \ TICK_{K_{ve}}(S_e, S_v, alias, K') \end{array}$$

where N_e is a nonce and K' is a symmetric key randomly generated by S_e. P_e is the public key of S_e. K_{ve} is a long-term key shared by S_v and S_e. In this attack, S_v will believe C_a is a subscriber of S_e but actually C_a is a subscriber of S_h.

Now suppose the fourth message flow is included. It allows S_v to send P_v to C_a for the purpose of future authentication. The message flow is denoted by $TICK_{K_{av}}(S_v, alias, S_v, P_v)$. We can see that in this attack, even S_e simply relays the fourth message, $TICK_{K'}(S_v, alias, S_v, P_v)$ directly from S_v to C_a, C_a will still accept, but get a wrong value of P_v.

B. Go-Kim Anonymous Authentication Protocol for Roaming

In [6], Go and Kim proposed another authentication protocol preserving user anonymity. Compared with Samfat et al.'s scheme, their protocol supports Server Authentication. However, we will show that their scheme does not support Subscription Validation.

We use the same set of notations as defined in Sec. IV-A. Below are some additional notations. Let G be a cyclic group generated by g of prime order q. Assume the discrete logarithm problem in G is hard. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be some cryptographically strong hash functions. Let $(\hat{S}_H, P_h) \in \mathbb{Z}_q \times G$ be S_h's private key/public key pair such that $P_h = g^{\hat{S}_H}$. Let $(\hat{S}_V, P_v) \in \mathbb{Z}_q \times G$ be S_v's private/public key pair such that $P_v = g^{\hat{s}_V}$. Let T_1, T_2 and T_3 be timestamps. The Go-Kim protocol is shown as follows.

$$\begin{array}{lcl} C_a & : & N_a \in_R \mathbb{Z}_q, \ K_{ah} = P_h^{N_a}, \\ & & alias = \mathcal{E}_{K_{ah}}(\mathcal{H}_1(C_a) \oplus g^{N_a}) \\ C_a \Rightarrow S_v & : & S_h, \ alias, \ g^{N_a} \\ S_v & : & N_v \in_R \mathbb{Z}_q \\ S_v \Rightarrow S_h & : & alias, \ g^{N_v}, \ g^{N_a}, \ Sig_V(g^{N_v}, g^{N_a}, alias, S_v), \ T_1 \\ S_h & : & N_h \in_R \mathbb{Z}_q, \ K_{hv} = \mathcal{H}_2(g^{N_v N_h}, P_v^{N_h}) \\ S_h \Rightarrow S_v & : & g^{N_h}, \ \mathcal{E}_{K_{hv}}(Sig_H(g^{N_h}, g^{N_v}, \mathcal{H}_1(C_a) \oplus g^{N_a}, S_h), \\ & & \qquad \mathcal{H}_1(C_a) \oplus g^{N_a}), \ T_2 \\ S_v & : & alias' = \mathcal{H}_1(g^{N_v N_a}, \mathcal{H}_1(C_a)), \\ & & K_{av} = \mathcal{H}_2(g^{N_v N_a}, g^{\hat{S}_V N_a}) \\ S_v \Rightarrow C_a & : & g^{N_v}, \ \mathcal{E}_{K_{av}}(\mathcal{H}_1(g^{N_v}, g^{N_a}, alias', S_v), T_2), \ T_3 \\ C_a \Rightarrow S_v & : & \mathcal{E}_{K_{av}}(Sig_A(g^{N_a}, g^{N_v}, T_2, S_v), T_3) \end{array}$$

Deposit-Case Attack: Direct application of the deposit-case attack is not obvious here. This is because the malicious server M has to decrypt $alias$ and obtain the real identity of C_a in order to deliver the correct value to S_v and let C_a accept when C_a receives a commitment of $alias'$ in the second last message flow. However, M does not know K_{ah} which is needed to decrypt $alias$.

Note that $alias$ is used to hide the real identity of C_a so that the Go-Kim protocol can provide user anonymity and untraceability against eavesdroppers. Hence before launching the deposit-case attack, M should find out the real identity of C_a. Below are the details on how M can find out C_a's real identity[3] and launch the deposit-case attack. Let $P_M \in G$ be M's public key.

[3]Precisely, M finds out the value of $\mathcal{H}_1(C_a)$ in the attack. However, the commitment $\mathcal{H}_1(C_a)$ has already provided enough information for an adversary to trace and reveal the identity of the user.

$$
\begin{array}{lcl}
C_a & : & N_a \in_R \mathbb{Z}_q,\ K_{ah} = P_h^{N_a}, \\
 & & alias = \mathcal{E}_{K_{ah}}(\mathcal{H}_1(C_a) \oplus g^{N_a}) \\
C_a \Rightarrow M & : & S_h,\ alias,\ g^{N_a} \\
M & : & N_1 \in_R \mathbb{Z}_q \\
M \Rightarrow S_h & : & alias,\ g^{N_1},\ g^{N_a},\ Sig_M(g^{N_1}, g^{N_a}, alias, M),\ T_0 \\
S_h & : & N_h \in_R \mathbb{Z}_q,\ K_{hm} = \mathcal{H}_2(g^{N_1 N_h}, P_M^{N_h}) \\
S_h \Rightarrow M & : & g^{N_h},\ \mathcal{E}_{K_{hm}}(Sig_H(g^{N_h}, g^{N_1}, \mathcal{H}_1(C_a) \oplus g^{N_a}, S_h), \\
 & & \mathcal{H}_1(C_a) \oplus g^{N_a}),\ T_2 \\
M \Rightarrow S_v & : & M,\ alias,\ g^{N_a} \\
S_v & : & N_v \in_R \mathbb{Z}_q \\
S_v \Rightarrow M & : & alias,\ g^{N_v},\ g^{N_a},\ Sig_V(g^{N_v}, g^{N_a}, alias, S_v),\ T_1 \\
M & : & N_2 \in_R \mathbb{Z}_q,\ K_{mv} = \mathcal{H}_2(g^{N_v N_2}, P_v^{N_2}) \\
M \Rightarrow S_v & : & g^{N_2},\ \mathcal{E}_{K_{mv}}(Sig_M(g^{N_2}, g^{N_v}, \mathcal{H}_1(C_a) \oplus g^{N_a}, M), \\
 & & \mathcal{H}_1(C_a) \oplus g^{N_a}),\ T_2 \\
S_v & : & alias' = \mathcal{H}_1(g^{N_v N_a}, \mathcal{H}_1(C_a)), \\
 & & K_{av} = \mathcal{H}_2(g^{N_v N_a}, g^{\hat{S}_V N_a}) \\
S_v \Rightarrow C_a & : & g^{N_v},\ \mathcal{E}_{K_{av}}(\mathcal{H}_1(g^{N_v}, g^{N_a}, alias', S_v), T_2),\ T_3 \\
C_a \Rightarrow S_v & : & \mathcal{E}_{K_{av}}(Sig_A(g^{N_a}, g^{N_v}, T_2, S_v), T_3)
\end{array}
$$

In the attack, the malicious server M first pretends to be a foreign server, contacts C_a's home server S_h, and claims that C_a is communicating with M. S_h then innocently sends C_a's real identity to M. After that, M launches the deposit-case attack by impersonating C_a and sending a modified message to S_v (illustrated as the first message from M to S_v in the diagram above). This message makes S_v believe that M is the home server of C_a while A believes that he has informed S_v that S_h is his home server.

Notice that C_a and S_v will still agree on the same key K_{av} when the attack completes. Hence the attack is carried out successfully and will not be discovered by any of the three honest parties.

V. The AAKE-R Scheme

We now propose a generic way to construct AAKE-R. In our construction, we use conventional authenticated key exchange protocols as building blocks. In the following, we first describe the building blocks.

A. Authenticated Key Exchange (AKE)

AKE is a two-party protocol which allows the parties to authenticate each other and simultaneously come into possession of a shared session key. There are many different kinds of AKE protocols in the literature, both symmetric and asymmetric. For the symmetric case [18], [19], the parties are assumed to have some long-term secret key (or password) shared. For the asymmetric case [16], [20], each of the parties is already associated with some public key pair.

To describe one protocol run of an AKE, we need to specify the inputs of the two parties and the generation of the session key. In the following, we introduce a notation to describe an AKE protocol run for the asymmetric case. Let the identities of the two parties be A and B. Let the public key pairs of A and B be $(\hat{s}_A, P_A)$ and $(\hat{s}_B, P_B)$, respectively. Suppose the session key generated is σ, the AKE protocol run is denoted by

$$\sigma \leftarrow AKE(A, B, (\hat{s}_A, P_A), (\hat{s}_B, P_B))$$

There is a special type of AKE schemes: authenticated key transport (AKT). The only difference between AKT and AKE is that the former has the session key prepared by one party and 'transported' securely to the other party. That is, one party has the exclusive key control [15]. By using the symbols above, suppose the session key σ is prepared by A and is transported to B, the AKT protocol run is denoted by

$$AKT(A, (\hat{s}_A, P_A)) \xrightarrow{\sigma} (B, (\hat{s}_B, P_B))$$

Anonymous Authenticated Key Exchange: Some conventional AKE schemes can be converted to support one-party anonymity against eavesdroppers. This type of schemes allows one of the two parties to hide its identity in the messages exchanged so that no eavesdropper can identify who that party is. Notice that these schemes are fundamentally different from AAKE-Rs. One major difference is that an AAKE-R scheme hides the user's identity from the foreign server while an AKE scheme which supports one-party anonymity against eavesdroppers does not hide the user's identity from the other communicating party.

An example of AKE schemes which equips this property has been discussed in [20, Sec. 9]. The approach is to have the party who requires anonymity against eavesdroppers to establish a secure session with the other party using some secure certificate-based one-way authenticated key establishment scheme (e.g. SSL/TLS [21]) and then on top of the secure channel, the anonymous party will identify and authenticate itself to the other party (e.g. showing a password). If the anonymous party also has a public key pair such that the other party knows the public key already, the anonymous party can also use the key pair to generate a signature as a response on a challenge made by the other party. Both challenge and response are transferred on top of the secure channel and they are also required to be fixed in length no matter what signing algorithm the anonymous party chooses and how big the key pair is.

Other AKE/AKT schemes that are believed to have this property include [22], [23], [24]. To denote that A is hiding its identity, we use the notation below.

$$\sigma \leftarrow AAKE(A, B, (\hat{s}_A, P_A), (\hat{s}_B, P_B))[A]$$

For simplicity, assume all the session keys mentioned above are k bits long. The requirements of anonymity and untraceability on AAKE should follow the definitions given in Sec. III-D in the straightforward way.

B. Protocol Description

As defined, the protocol consists of three entities: user C_a, home server S_h such that $f(C_a) = S_h$ and foreign server S_v where $S_h \neq S_v$. Assume that there is a direct link between C_a and S_v and another direct link between S_v and S_h. But there is no direct link between C_a and S_h. For all communications between C_a and S_h, messages are relayed by S_v.

Let $(\hat{s}_v, P_v)$ be the public key pair of S_v for some public key encryption scheme $\mathcal{E}$. Assume that S_v broadcasts P_v associated with a certificate and C_a has obtained P_v and checked its validity using the associated certificate before running the protocol. We use $\mathcal{E}_{P_v}$ to denote the encryption under the public key P_v. Let $(\hat{s}_a, P_a)$ and $(\hat{s}_h, P_h)$ be the public key pairs of C_a and S_h, respectively. Assume that for each server $S_h \in \mathcal{S}(k)$, the public key P_h of S_h is known to

all users in $f^{-1}(S_h)$ and also for each user $C_a \in \mathcal{C}(k)$, the public key P_a of C_a is known to $f(C_a)$. In other words, we assume that each user knows its home server's public key and each server knows the public keys of all its subscribers. In practice, the user and his home server can send their public keys to each other when the user subscribes to the server in some registration phase. For simplicity, we also assume that all servers know the public keys of all other servers in $\mathcal{S}(k)$. In practice, this can be replaced by a certificate-based solution for providing scalability.

Let H_1, H_2 and H_3 be cryptographically strong hash functions. Each of them maps from $\{0,1\}^*$ into $\{0,1\}^k$. Below is the protocol description.

1) C_a randomly generates $k_a \in_R \{0,1\}^k$ and sends $m_1 = \mathcal{E}_{P_v}(S_h \| H_1(k_a))$ to S_v where '$\|$' denotes string concatenation.
2) S_v decrypts m_1 using the private key $\hat{s}_v$ and separates it into two halves: the first k-bit binary string is S_h, and the second k-bit binary string should be $H_1(k_a)$. Here we denote it by α. It then 'informs' (by sending a prespecified message) S_h that there is a user who claims to be its subscriber. This message will be included in the next step for saving one round of communication.
3) C_a and S_h start up an AAKE run via S_v and attain
$$c \leftarrow AAKE(C_a, S_h, (\hat{s}_a, P_a), (\hat{s}_h, P_h))[C_a]$$
if $f(C_a) = S_h$. Otherwise, both entities halt with failure. S_v will also halt with failure after being informed by C_a or S_h or timeout.
4) S_h computes $\Pi = H_2(C_a, S_h, S_v, c)$.
5) S_h and S_v start up an AKT run and attain
$$AKT(S_h, (\hat{s}_h, P_h)) \overset{\Pi}{\rightarrow} (S_v, (\hat{s}_v, P_v)).$$
If the AKT fails to complete, both entities halt with failure. C_a will also halt with failure after being informed by S_v or timeout.
6) S_v randomly generates $k_b \in_R \{0,1\}^k$ and sends $m_2 = \alpha \oplus k_b$ to C_a.
7) C_a obtains k_b as $m_2 \oplus H_1(k_a)$ and sends $m_3 = H_1(k_b) \oplus k_a$ to S_v.
8) S_v obtains k_a from m_3 and checks if $H_1(k_a) = \alpha$. If it is true, continue. Otherwise, S_v rejects the connection and halts.
9) Each of C_a and S_v computes the session key σ as $H_2(S_h, S_v, k_a, k_b, \Pi)$ and jointly conduct the following key confirmation steps.
   a) S_v sends $m_4 = H_3(S_h, S_v, H_1(k_a), k_b, \Pi)$ to C_a.
   b) C_a checks if $m_4 \overset{?}{=} H_3(S_h, S_v, H_1(k_a), k_b, \Pi)$. If it is true, C_a sends $m_5 = H_3(S_h, S_v, k_a, k_b, \Pi)$ back to S_v and accepts the connection. Otherwise, C_a rejects and halts.
   c) S_v checks if $m_5 \overset{?}{=} H_3(S_h, S_v, k_a, k_b, \Pi)$. If it is correct, S_v considers that the connection is established. Otherwise, S_v halts with failure.

   The messages can be piggybacked in the last two message flows.
10) Both C_a and S_v destroy their copies of k_a and k_b after accepting the connection.

The optimized protocol after piggybacking the last two message flows is illustrated in Fig. 1.

C. Security Analysis

In the following, we assume that all hash functions used in our scheme behave like random oracles [25] (random functions). Server authentication is done by the following challenge-response pair,

$$(\mathcal{E}_{P_v}(S_h || H_1(k_a)),\ H_3(S_h, S_v, H_1(k_a), k_b, \Pi)).$$

Only S_v, who has $\hat{s}_v$, can obtain the value of $H_1(k_a)$ from the first item of the pair, and the response containing the digest of $H_1(k_a)$ lets C_a authenticate S_v.

Subscription validation is done in three steps. First, S_h is involved to authenticate C_a. This is done by using the AAKE mechanism. Second, S_h sends S_v a *credential*, which comprises the entire transcript of the AKT protocol run, for testifying that the user who has involved in the AAKE in the first step is a subscriber of S_h. Third, S_v ensures that the user communicating with S_h in the first step is also the one who is currently communicating with. This is done by having the user send the last message component, $H_3(S_h, S_v, k_a, k_b, \Pi)$ to S_v. Since besides S_h and S_v, only the user who has communicated with S_h in the first step can compute the value of Π[4].

For key establishment, we will show that only C_a and S_v are sharing the fresh session key after one protocol run. First, only C_a, S_v and S_h know the value of Π and therefore only these three parties are able to compute the session key $\sigma = H_2(S_h, S_v, k_a, k_b, \Pi)$ if they also know k_a and k_b. As in the subscription validation, we exclude the scenario that S_h is impersonating its own subscriber. So in the following, we only need to show that S_h cannot obtain at least k_b from the transcript of one protocol run. Notice that both $H_1(k_a) \oplus k_b$ and $H_1(k_b) \oplus k_a$ do not help get k_b since $H_1(k_a)$ and $H_1(k_b)$ are some unknown pseudorandom strings and no any bit information of k_a or k_b can be obtained from them. In addition, the first message flow does not leak any information of $H_1(k_a)$ provided that the underlying encryption function $\mathcal{E}$ is semantically secure against adaptive chosen ciphertext attacks [26]. Hence S_h cannot obtain session key σ from the transcript of the protocol run. On key control, it can be seen that joint key control is achieved and no party can predetermine the value of the session key when generating their session key component. The technique is the same as the commitment approach due to [15].

To show that the scheme is user anonymous, we first allow an adversary to choose a home server S_h from $\mathcal{S}(k)$ for some sufficiently large k, and also a set of $Q_3(k)$ subscribers in $f^{-1}(S_h)$. We then invoke the protocol by randomly choose a server from $\mathcal{S}(k) \setminus \{S_h\}$ as the foreign server S_v for that protocol run. We record the entire transcript. This is repeated until we obtain a set of $Q_3(k)$ transcripts / protocol runs. Denote the set by $\mathcal{T}$. We then randomly pick a user $C_a \in f^{-1}(S_h)$ and generate a transcript T_a. After invoking

[4]Note that S_h can always communicate with S_v and claim to be a subscriber of itself. This is reasonable in practice as S_h can always create new subscribers at its own will.

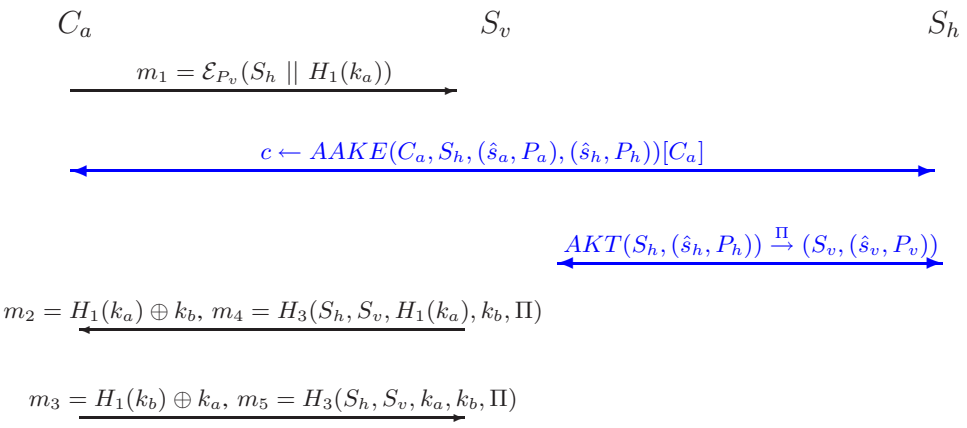

Fig. 1. The AAKE-R Protocol

the adversary, we wait for the adversary to return its guess of the user's identity.

In the protocol, we can see that besides AAKE and the value of Π, there is no information related to the identity of C_a. Without knowing C_a and c, which is the secret output of AAKE but not known to the adversary, Π is just the digest of two unknown values and does not help the adversary obtain any additional information of C_a. Therefore, the degree of user anonymity of the protocol reduces to that of the AAKE scheme. Similarly, user untraceability is also ensured by the security assumption of the underlying AAKE scheme.

An eavesdropper can find out the identity of the home server from the transcript of one protocol run by simply looking into the portion of the transcript corresponding to AAKE and AKT. On the other hand, it becomes easy to provide home server hiding when considering eavesdroppers who can only access messages between C_a and S_v but not between S_v and S_h. In this adversary model, our scheme also provide identity hiding for the home server by building a secure channel between C_a and S_v and carrying out the AAKE on top of the secure channel. However under the security definition we give in Sec. III-D, the eavesdroppers are more versatile than that.

D. Instantiation and Performance Evaluation

In Sec. V-B, we propose a generic construction of AAKE-R protocols. The AAKE and AKT protocols can be instantiated by concrete protocols, and optimization of message flows can be applied so that the instantiation can be carried out efficiently for low-power wireless roaming. Suppose we have a 3-round AAKE protocol as follows [5]:

$$C_a \rightarrow S_h : M_1$$
$$C_a \leftarrow S_h : M_2$$
$$C_a \rightarrow S_h : M_3$$

Suppose the session key σ can be computed by S_h once after M_1 is received. The AAKE-R generic construction described in Sec. V-B can be optimized by using piggybacking technique shown in Fig. 2.

1) C_a sends m_1 and the first message (denoted by M_1) of the AAKE protocol to S_v.
2) Upon receipt of the message (m_1, M_1) from C_a, S_v decrypts m_1 using the private key $\hat{s}_v$ and separates it into two halves: the first k-bit binary string is S_h, and the second k-bit binary string should be $H_1(k_a)$. Here we denote it by α. It then forwards M_1 to S_h.
3) After receiving M_1 from S_v, S_h computes c and then $\Pi = H_2(C_a, S_h, S_v, c)$. S_h and S_v then start up an AKT run (e.g. [23]) to attain
$$AKT(S_h, (\hat{s}_h, P_h)) \overset{\Pi}{\rightarrow} (S_v, (\hat{s}_v, P_v)).$$
If the AKT fails to complete, both entities halt with failure. C_a will also halt with failure after being informed by S_v or timeout. S_h also sends M_2 to S_v[6].
4) S_v randomly generates $k_b \in_R \{0,1\}^k$ and sends $m_2 = \alpha \oplus k_b$, $m_4 = H_3(S_h, S_v, H_1(k_a), k_b, \Pi)$ and M_2 to C_a.
5) C_a obtains k_b as $m_2 \oplus H_1(k_a)$ and verifies m_4 and M_3. If the verification succeeds, C_a sends $m_3 = H_1(k_b) \oplus k_a$ together with $m_5 = H_3(S_h, S_v, k_a, k_b, \Pi)$ and M_3 to S_v, C_a accepts the connection and computes the session key σ as $H_2(S_h, S_v, k_a, k_b, \Pi)$. Otherwise, C_a rejects and halts.
6) S_v obtains k_a from m_3 so that S_v can check if $H_1(k_a) = \alpha$ and $m_5 = H_3(S_h, S_v, k_a, k_b, \Pi)$. If the verification succeeds, continue. Otherwise, S_v rejects the connection and halts. S_v also forwards M_3 to S_h.
7) S_h verifies message M_3, and informs S_v the verification result by sending an acknowledgement ACK to S_v in an authenticated way. This is feasible as S_h and S_v are sharing a secret session key Π. For example, this can be done as follows: if the verification of M_3 succeeds, $ACK = H_3('1', \Pi)$; otherwise, $ACK = H_3('0', \Pi)$.
8) S_v verifies ACK, if $ACK = H_3('0', \Pi)$, S_v halts with failure. Otherwise, S_v accepts the connection and computes session key σ as $H_2(S_h, S_v, k_a, k_b, \Pi)$.
9) Both C_a and S_v destroy their copies of k_a and k_b after accepting the connection.

Performance: In the real world, servers are usually connected by communication links with much higher bandwidth than the links between roaming users and servers. Due to

[5]We refer readers to [24] for a concrete AAKE protocol.

[6]This message flow can be combined with the last message flow of the AKT protocol

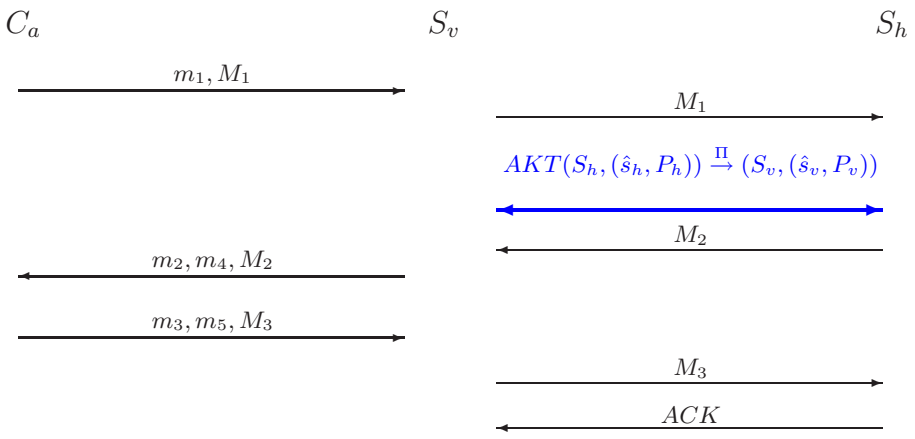

Fig. 2. The Optimized AAKE-R Protocol

the limited wireless spectrum provided for communications between roaming users and servers and limited battery power of roaming users, the performance of a roaming protocol is mainly determined by the communication rounds between the mobile roaming user and the foreign server. In the protocol instantiation above, there are only three communication rounds between the mobile user and the foreign server, and this is commonly believed to be the minimum number of rounds required in order to achieve joint key control.

In the instantiation above, the AAKE protocol in [24] performs a Diffie-Hellman key exchange, a digital signature verification and a digital signature generation. Diffie-Hellman key exchange protocol has already been used in existing cellular networks (e.g. CDPD [27] in North America). The computational complexity of digital signature verification relies a lot on the public key. In the case of the low exponent RSA algorithm (e.g. e = 3), the signature verification takes only two modular multiplications. And for the digital signature scheme used by the mobile user, we can choose Schnorr's signature scheme [28] where all the users registered in the same home domain can use the same multiplicative group. The reason is that Schnorr's signature scheme supports precomputation, only one hash function and one modular multiplication is needed in real time signature generation. And for the encryption under the foreign server's public key, we can choose low exponent RSA algorithm again, which takes another two modular multiplication operations.

In Table II, we make a comparison among our protocol instantiation and some related protocols in terms of both security and performance.

As we can see in the table, the roaming protocol proposed by Hwang et al. [14] may provide server authentication and subscription validation, but under the assumption that the roaming user also checks the identity information in the message from the foreign server S_v to the roaming user C_a. However in [14], this assumption is not made. And besides Hwang et al.'s protocol and ours, none of the remaining protocols is secure against deposit-case attack.

On key establishment, only our protocol and Go-Kim protocol support session key establishment while preventing the home server from obtaining the key. As explained previously in this paper, this property is important for roaming applications. Besides this feature, only our protocol supports joint key control.

On user anonymity, GSM and 3GPP roaming protocols provide a certain degree of anonymity by using some temporary identity called TMSI (Temporary Mobile Subscriber Identity) rather than the real identity IMSI (International Mobile Subscriber Identity) for each roaming user. However, this mechanism does not help prevent those visited foreign servers from tracing a roaming user when the user hops from one foreign server to another. Different sessions of the same user inside one foreign domain can also be easily linked by the foreign server. We refer to the security of this type as Insider/(Visited server) security and this level of security is only supported by our protocol.

VI. Concluding Remarks

In this paper, we focus on keeping the identity of a user secret and untraceable from foreign servers, eavesdroppers and other users while allowing the user to conduct authenticated key exchange with the serving foreign server. On the other side, the home server can still keep track the user's whereabouts for the purpose of billing, providing customized services or compelling security policies.

As an extension of the anonymous roaming concept, we can further consider a scenario in which each user is subscribed to an independent agent who is responsible for charging the user for accessing a server and clearing the bill sent by the server. The agent itself is not a server. Hence one can consider each user in the system to be always roaming at some foreign network. When a user is requesting for connection to a server, the user only needs to let the network know who its agent is and show that it is a legitimate subscriber of the agent. We target to make sure that the real identity of the user is not given and connection requests from the same user should not be linked.

This extension of separating the roles of subscription management and networking service provision may benefit both users and service providers. For users, they have the flexibility of choosing the serving servers. For servers, they can focus on improving their service quality and will provide services to potentially a larger set of customers. We believe that this is a

TABLE II
COMPARISON AMONG ROAMING PROTOCOLS

	GSM	3GPP	SMA [5]	Go-Kim [6]	Hwang-Chang [14]	Our Protocol
Server Authentication	¬	✓	¬	✓	?	✓
Subscription Validation	P	P	P	P	?	✓
Key Establishment	E	E	E	EH	E	EHJ
User Anonymity	P	P	✓	A [11]	¬	✓
User Untraceability	O	O	O	A [11]	¬	OI
$C_a \leftrightarrow S_v$	4	3	2	3	3	3

Notations:
✓ Satisfied
P Partially satisfied
E Key establishment
H Home server does not know the session key
J Joint key control
O Outsider security
I Insider/(Visited server) security
? Depending on the assumption
¬ Not considered
A Attack exists

plausible trend of tomorrow's roaming networks. In addition, with the agent, there is no home server and it is not needed to concern about the leaking of the identity of home servers.

We defined five security requirements for an anonymous and authenticated key exchange protocol for roaming (AAKE-R). They are Server Authentication, Subscription Validation, Key Establishment, User Anonymity and User Untraceability. Among the five requirements, Subscription Validation is the most difficult one to achieve. We also proposed a new practical attack called deposit-case attack. We showed that Samfat, et al.'s protocol [5] and the Go-Kim anonymous roaming protocol [6] cannot provide subscription validation due to this attack. An AAKE-R scheme can be a three-party protocol which requires the home server of the roaming user to get involved (online case), or a two-party one which does not requires the home server to get involved (offline case). We stress that the security requirements of these two types of schemes should be the same.

By using provably secure authenticated two-party key exchange protocols as building blocks, we proposed a secure and generic AAKE-R construction. Our construction does not use alias mechanism for achieving user anonymity and untraceability. Hence it does not have the alias synchronization issue. Other merits of our protocol include the support of joint key control, and not relying on any special security assumptions on the link between the visiting server and the user's home server. The protocol can also be instantiated efficiently. By piggybacking some message flows, the number of message flows between the roaming user and the visiting server is only three and all operations can be implemented efficiently.

Finally, we advocate the concept of role division [20]. We do not make any assumption on the high-level protocols. An AAKE-R scheme should just achieve the security goals defined which are solely for creating a secure channel between the anonymous roaming user and the foreign server. For other application-oriented requirements, such as billing and compelling security policies, they should be taken care of by high-level protocols. We consider the study of these additional requirements as our future work.

ACKNOWLEDGEMENTS

We would like to thank Xi Zhang, Yuguang Fang and all the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications*. Published by the authors, 1992.
[2] *Mobile Station-Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems (TIA/EIA-95-B-99)*, The Telecommunications Industry Association (TIA), Feb 1999.
[3] *3GPP TS 33.102: 3rd Generation Partnership Project 3GPP, 3G Security, Security Architecture*, Technical Specification Group (TSG) SA, Oct 2003.
[4] G. Ateniese, A. Herzberg, H. Krawczyk, and G. Tsudik, "On traveling incognito," in *Proc. IEEE Workshop on Mobile Systems and Applications*, Dec 1994.
[5] D. Samfat, R. Molva, and N. Asokan, "Untraceability in mobile networks," in *Proc. MobiCom '95*, pp. 26–36.
[6] J. Go and K. Kim, "Wireless authentication protocol preserving user anonymity," in *Proc. 2001 Symposium on Cryptography and Information Security (SCIS 2001)*, Jan. 2001, pp. 159–164.
[7] V. Varadharajan and Y. Mu, "Preserving privacy in mobile communications: A hybrid method," in *Proc. IEEE International Conference on Personal Wireless Communications 1997*, pp. 532–536.
[8] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. of the ACM*, vol. 24, pp. 84–88, Feb. 1981.
[9] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocations," in *Proc. EUROCRYPT 2001*. Springer-Verlag, 2001, pp. 93–118, lNCS 2045.
[10] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.
[11] D. Wong, "Security analysis of two anonymous authentication protocols for distributed wireless networks," in *Proc. 3rd IEEE Intl. Conf. on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)*. IEEE Computer Society, Mar. 2005, pp. 284–288.

[12] G. Yang, D. Wong, and X. Deng, "Deposit-case attack against secure roaming," in *Information Security and Privacy, 10th Australasian Conference, ACISP 2005*. Springer-Verlag, 2005, pp. 417–428, lNCS 3574.
[13] L. Buttyan, C. Gbaguidi, S. Staamann, and U. Wilhelm, "Extensions to an authentication technique proposed for the global mobility network," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 373–376, Mar. 2000.
[14] K. F. Hwang and C. C. Chang, "A self-encryption mechanism for authentication of roaming and teleconference services," *IEEE Trans. Wireless Commun.*, vol. 2, no. 2, pp. 400–407, Mar. 2003.
[15] C. Mitchell, M. Ward, and P. Wilson, "On key control in key agreement protocols," *Electron. Lett.*, vol. 34, pp. 980–981, 1998.
[16] W. Diffie, P. C. V. Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes, and Cryptography*, vol. 2, no. 2, pp. 107–125, June 1992.
[17] *Announcing the Advanced Encryption Standard (AES)*, NIST FIPS PUB 197, Nov. 2001.
[18] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. CRYPTO 93*. Springer-Verlag, 1994, pp. 232–249, lNCS 773.
[19] IEEE, *P1363.2 / D15: Standard Specifications for Password-based Public Key Cryptographic Techniques*, May 2004.
[20] V. Shoup, "On formal models for secure key exchange," http://www.shoup.net, Tech. Rep. 1999 revision of IBM Research Report RZ 3120, Nov 1999.
[21] T. Dierks and C. Allen, *RFC 2246: The TLS Protocol Version 1.0*, IETF RFC 2246, Jan 1999.
[22] G. Horn and B. Preneel, "Authentication and payment in future mobile systems," in *Proc. ESORICS'98*, pp. 277–293.
[23] C. Boyd and D. Park, "Public key protocols for wireless communications," in *Proc. 1st International Conference on Information Secuirty and Cryptology (ICISC'98)*, pp. 47–57.
[24] G. Yang, D. Wong, X. Deng, and H. Wang, "Anonymous signature schemes," in in *Proc. 9th International Workshop on Practice and Theory in Public Key Cryptography PKC 2006*. Springer, 2006, pp. 347–363, lNCS 3958.
[25] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. First ACM Conference on Computer and Communications Security*. Fairfax: ACM, 1993, pp. 62–73.
[26] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Proc. CRYPTO 91*. Springer, 1992, pp. 433–444, lNCS 576.
[27] *Cellular Digital Packet Data (CDPD) System Specification*, July 1993, release 1.0.
[28] C. P. Schnorr, "Efficient signature generation by smart cards," *J. of Cryptology*, vol. 4, no. 3, 1991.

Guomin Yang received his bachelor and master degrees from the Department of Computer Science, City University of Hong Kong, in 2004 and 2006 respectively. Currently he is a Ph.D. candidate in the same department, under the supervision of Prof. Xiaotie Deng and Dr. Duncan S. Wong. Mr. Yang's research interests include cryptography and communication security.

Duncan S. Wong received the B.Eng. degree from the University of Hong Kong in 1994, the M.Phil. degree from the Chinese University of Hong Kong in 1998, and the Ph.D. degree from Northeastern University, Boston, MA, U.S.A. in 2002. He is an assistant professor in the Department of Computer Science at the City University of Hong Kong.

Xiaotie Deng got his B. Sci. from Tsinghua University, Beijing, China, in 1982, and his M. Sci. at Chinese Academy of Sciences, Beijing, China in 1984, his Ph.D. at Stanford University, California, USA, 1989. After finishing PhD, he received an International Research Fellowship from Natural Science and Engineering Council of Canada to do postdoctoral research at Simon Fraser University, at British Columbia, Canada. In 1991, he joined York University, Toronto, as an assistant professor, and then tenured as an associate professor. In 1997, he joined City University of Hong Kong. He is now a full professor.