Handauth: Efficient Handover Authentication with Conditional Privacy for Wireless Networks

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

Abstract—Existing mechanisms for handover authentication mainly focus on designing a secure authentication module, little attention has been paid to protect users' privacy when they are authenticated by the access points for data access. Further, most existing approaches do not support user revocation. In this paper, we present a secure and efficient authentication protocol named *Handauth*. Similar to the mechanisms of this field, Handauth provides user authentication and session key establishment. However, compared to other well-known approaches, Handauth not only enjoys both computation and communication efficiency, but also achieves strong user anonymity and untraceablility, forward secure user revocation, conditional privacy-preservation, AAA server anonymity, access service expiration management, access point authentication, easily scheduled revocation, dynamic user revocation and attack resistance. Experimental results show that the proposed approach is feasible for real applications.

Index Terms—Handover authentication, privacy, revocation, wireless networks

1 Introduction

Nowadays, various wireless networks such as telecommunication systems, roadside-to-vehicle communication systems and WLANs have become widely available and interconnected. To provide seamless access services for mobile users (MUs) (e.g., PDA, laptop computer, smart phone and vehicle) without being limited by the geographical coverage of each access point, handover authentication modules have been deployed. Regardless of the technology implemented, as shown in Fig. 1, a typical handover authentication scenario involves three parties: mobile users, access points (APs) and an Authentication, Authorization, and Accounting (AAA) server. Before entering the network, an MU selects an AAA server for registration, then subscribes services and connects to an AP for accessing data. When the MU moves from the current AP (i.e., $AP1$) into a new AP (i.e., $AP2$), handover authentication should be performed at $AP2$. Here, the two circles indicate the transmission ranges of $AP1$ and $AP2$, respectively. Through handover authentication, $AP2$ authenticates the MU to protect itself from illegitimate access. At the same time, a session key should be established between the MU and $AP2$ to protect the user's data against attacks.

Privacy is a serious concern for the above handover authentication services whereas mobile privacy protection is a complicated issue. Users are deeply concerned about their privacy-related information such as the identity, position, and roaming route. Unfortunately, in current handover authentication techniques [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], it is commonly assumed that the APs are trustworthy and would keep users' privacy-related information confidential. However, since such information is extremely sensitive and coveted by many companies, which may use it to improve their business, such an assumption may not be valid. Therefore, a user should be protected from the prying eyes of APs. Without appropriate security and privacy guarantees, users are reluctant to accept such mobile services. To satisfy the security and privacy requirements, it is prerequisite to elaborately design an efficient handover authentication mechanism to achieve security and privacy preservation for practical wireless networks.

A secure and efficient handover authentication protocol should satisfy the following requirements:

1. **User authentication.**
2. **Session key establishment.**
3. **Low communication cost and computation complexity.** In general, an MU does not have sufficient resources in comparison with fixed nodes such as APs. Therefore, a handover authentication process should minimize energy consumption of MUs. Additionally, such a process should be fast enough to maintain persistent connectivity.
4. **Strong user anonymity and untraceablility.** It allows an MU not to expose its private information to eavesdroppers or APs.
5. **Provision of user revocation mechanism with forward secrecy.** Due to some reasons (e.g., the subscription period of a user has expired or a user's secret key has been compromised), handover authentication should allow an AP to find out whether an MU is revoked. At the same time, however, it should also guarantee the anonymity of the revoked user's protocol runs before the revocation, which means forward secure user revocation.
6. **Conditional privacy preservation.** Although it is desirable to provide strong user anonymity and untraceablility, it is the liability for the AAA server to reveal the related private information (e.g., identity, position) of a user in case of emergency (e.g., enhanced 911 location service mandated by US Federal Communications Commission).
7. **AAA server anonymity.** Besides the identity of the MU, the identity of its AAA server should also be hidden from eavesdroppers and the legitimate network entities except the visited AP [17]. Otherwise, the real identity of an MU may be discovered by analyzing the traffic between a visited AP and its AAA server. In other words, each time when a user accesses the network, if the identity of its AAA server is not protected, information about the user's real identity may be inferred. This is illustrated by the following example. If an aliased user x visiting a remote access point $AP2$ in Germany wants to authenticate to its AAA server *WhiteHouse.Gov* and an adversary happens to know that the only user from *WhiteHouse.Gov* currently in Germany is *President@WhiteHouse.Gov*, the adversary can conclude that x in fact corresponds to *President@WhiteHouse.Gov*.
8. **Local access service expiration.** With the involvement of the AAA server, each MU should be permitted to access the services only during its subscription period. For example, in mobile phone services, it is necessary for the AAA server to precisely control the service time of an MU according to service payments and managements.
9. **Local AP validation.** Most handover authentication schemes just consider the authentication of MUs by the visited AP. However, it is also important that each MU is able to verify that the visited AP is authorized by the AAA server to offer access services without the help of its AAA server. Otherwise, an imitated AP will easily obtain the private information of the MUs who carelessly connect to it. We consider data phishing attack as an example. In such

- *D. He, J. Bu, and C. Chen are with the College of Computer Science, Zhejiang University, Hangzhou 310027, P.R. China. E-mail: hedaojinghit@gmail.com.*
- *S. Chan is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. E-mail: eeschan@cityu.edu.hk.*

Manuscript received 5 Apr. 2011; revised 3 Oct. 2011; accepted 9 Dec. 2011; published online 27 Dec. 2011.
Recommended for acceptance by S. Olariu.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2011-04-0216.
Digital Object Identifier no. 10.1109/TC.2011.258.

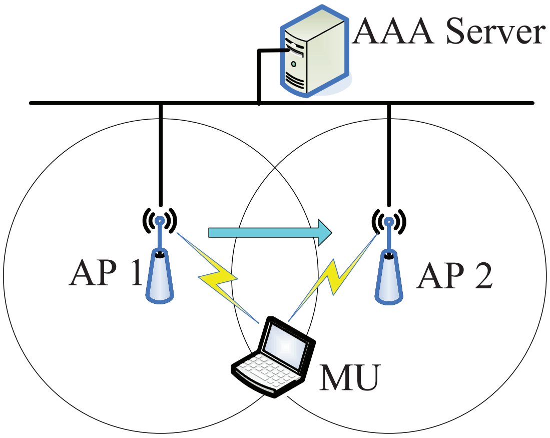

Fig. 1. Handover authentication overview.

an attack, an adversary may set up bogus APs and try to phish user connections to such APs. In this way, adversaries could control network connection and analyze users' data traffic for their benefits.

10. **Easily scheduled revocation.** To be more practical, it should easily allow a scheduled revocation after which a user will resume the services without reregistering to the AAA server. For example, a user may plan to suspend the services for a few months.
11. **Provision of dynamic user revocation mechanism.** Due to some reasons (e.g., a user's secret key has been compromised or a user has misbehaved), revocation of misbehaving users should take place at any time to prevent these users from jeopardizing the safety of other users and the network provider. Note that different from Requirements 8 and 10, dynamic user revocation occurs before the subscription period of a user expires.
12. **Attack resistance.** Clearly, handover authentication protocol should have ability to resist various kinds of attacks (e.g., Denial-of-Service (DoS) attacks).

Obviously, designing a secure and efficient handover authentication protocol is a nontrivial task because wireless networks are vulnerable to attacks and mobile users are resource-constrained. While Requirements 1-3 have been well addressed in the literature, to the best of our knowledge, Requirements 4-12 have been largely neglected. More importantly, when considering this research issue, we observe that none of the existing privacy-aware cryptographic primitives can be directly applied to achieve the goal discussed above. The detailed analysis to arrive at these conclusions will be given in Sections 2 and 3.1. This becomes a more severe issue given the trend that more and more wireless networks are being deployed. Motivated by this observation, this paper makes three main contributions:

1. We first identify the characteristics of handover authentication and then present a comprehensive set of requirements for the protocols of this kind. We show some security weaknesses and efficiency problems of current handover authentication protocols [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16].
2. We propose a novel approach to ensure secure and efficient handover authentication, called *Handauth*, which is built on the most efficient forward secure revocable group signature (FSR-GS) technique. Since FSR-GS was not originally designed for handover authentication protocols, a direct application of the method simply cannot satisfy Requirements 2-4 and 6-12, which are very challenging for ensuring secure, efficient, and robust access services for mobile users. For example, due to the sole network layer protocol design, it cannot preserve user anonymity and untraceability. Also, it does not provide Handauth with easily scheduled revocation, access service expiration, AAA server anonymity properties. To address these issues, some additional mechanisms are incorporated into the design of the proposed protocol. Finally, Handauth satisfies all of the above requirements. Moreover, Handauth can achieve scalability; it is efficient even in a large-scale network with many subscribers and many revoked users. Furthermore, it supports dynamic participation. New users can easily join the network, and users can easily be revoked when their subscriptions expire. Another desirable feature of Handauth is that each handover authentication does not involve the AAA server. APs only notify the AAA server of the authentication result after performing the handover authentication; thus, no extra delay is incurred in the authentication process.
3. In addition to the security analysis which demonstrates that Handauth indeed enforces its security guarantees, this paper also reports the experimental results of Handauth, showing its efficiency in practice.

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 and efficiency problems. Section 3 discusses the limitations of various existing privacy-aware cryptographic primitives and then introduces the most suitable one—the FSR-GS technique. Section 4 describes Handauth in detail. Then, in Section 5, we discuss some important issues about our protocol and further improve it. The simple proof and formal analysis of the security properties of Handauth are provided in Section 6. Experimental results and performance analysis are given in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

Due to the importance of handover authentication, many secure mechanisms [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] have been proposed for this purpose. However, most of existing solutions focus on Requirements 1-3. In telecommunication systems, the GSM [1] communication system are intended to provide user privacy by using a temporary identity called Temporary Mobile Subscriber Identity (TMSI) to identify an MU. However, a user's real identity called International Mobile Subscriber Identity (IMSI) is sent to the visited AP over the air in plaintext during the authentication process; thus, eavesdroppers over the radio network can easily identify the subscriber by its IMSI. Obviously, GSM cannot satisfy Requirement 4. The third-generation mobile cellular communication system UMTS [2], though enhanced from GSM, uses the same mechanism to provide anonymity for MUs. That is, UMTS also uses IMSI for the first registration at the visited AP, and obtains some TMSIs for subsequent sessions. Likewise, UMTS cannot achieve Requirement 4.

For securing WLAN roaming, the IEEE 802.1x standard employs Extensible Authentication Protocols (EAP), on which any authentication mechanism can be used for authenticating both the user and the network. In the EAP framework, some authentication methods including Message Digest 5 (MD5, IETF RFC 1321), Transport Layer Security (TLS, IETF RFC 2716), Tunneled TLS [3] and Protected Extensible Authentication Protocol [4] have been proposed. EAP-MD5 is primarily based on a one-way hash function. When using EAP-MD5, a subscriber computes the hash value with the password as input, the hash is transmitted over air to the visited AP for subscriber validation. Thus, EAP-MD5 cannot achieve user untraceability and AP authentication.

Additionally, in the other three EAP authentication methods, an MU performs mutual authentication with the AAA server through exchanging certificate with each other. Therefore, they cannot meet Requirement 4.

Additionally, some authentication schemes [5], [6], [7], [13], [14], [15], [16] with robust security features have been suggested. The approaches in [7], [13] make use of the simple operations such as one-way hash functions and exclusive-OR to achieve security goals. We observe that these work ([7], [13]) just focus on designing lightweight user authentication, but do not pay attention to provide strong security (e.g., Requirement 4). In [14], the focus is mainly on preventing from smart card breach, which is based on symmetric and public key cryptography. Also, the protocol of [16] makes use of symmetric cryptographic and hash operation primitives for secure authentication.

Moreover, all above techniques [5], [6], [7], [13], [14], [15], [16] require the AAA server to be involved in each protocol run. Such an approach suffers from security weaknesses and efficiency problems which include: 1) A communication round between the visited AP and the AAA server is required. When the AAA server is many hops away from the visited AP, this communication delay is even more crucial. 2) Since these protocols require a visited AP to unconditionally forward any login request, valid or invalid, to the AAA server, an adversary can easily launch DoS attacks on the AAA server through an AP. Thus, they cannot meet Requirement 12.

To solve the above issue, some approaches without involving the AAA server have been proposed in [8], [9], [10], [11], [12]. In order to enable an AP to locally check the validity of MUs, some complex cryptography techniques (e.g., ID-based cryptosystem, credentials based on chameleon hashing) are usually used. These methods of [8], [9] do not require the involvement of the AAA server, but we observe that these works do not consider Requirements 4-12. Recently, a handover authentication using the ID-based cryptosystem is presented in [10]. It performs a fast mutual authentication between an MU and an AP without involving the AAA server. Unfortunately, it suffers from the key escrow problem since the private key generator issues the private key of each MU. Also, the method cannot achieve user anonymity and conditional privacy protection. Later, a handover authentication scheme using credentials based on chameleon hashing has been proposed in [11]. The scheme provides robust key exchange and efficient authentication procedure. However, it cannot achieve user anonymity since a user always needs to send the same credential to an AP for verification. Additionally, a fast and efficient handover authentication in Vehicle-to-Infrastructure networks has been introduced in [12]. Due to the use of pseudo identity, however, it cannot achieve user untraceablility.

3 The Cryptographic Primitive of Handauth

3.1 Available Choices

We observe that none of the existing privacy-aware cryptographic primitives (e.g., standard digital signature, blind signature, group signature, ring signature, and extended techniques) can be directly applied to achieve the goal discussed above. Standard digital signature schemes will directly reveal an MU's identity. Blind signature and ring signature algorithms can only provide irrevocable anonymity, while here it demands conditional privacy preservation and hence revocable anonymity.

A viable approach is for each user to send a login request to the AP through a basic group signature technique. A basic group signature scheme allows one member of the group to sign a message such that any verifier can just verify if the message is originated from a group member without knowing the identity of the actual sender. Only the group manager can lift the anonymity of a signature and reveal the identity of the signer who created it. Group membership is controlled by the group manager, who generates the group's public key and provides individual members with their secret signing keys. To further support user revocation with forward secrecy, the group manager has to change and redistribute the group public key and secret keys of all but the revoked users. Therefore, it incurs enormous loads to nonrevoked users.

A more suitable approach is to use forward secure revocable group signature technique. Forward secure revocation allows a revoked group member to preserve the anonymity of its signatures generated before the revocation. However, we observe that although FSR-GS techniques have been proposed by researchers for a long time, most of the existing FSR-GS schemes (e.g., [18], [19], [20]) are not suitable for the construction of efficient handover authentication. These techniques (e.g.,[18], [19]) either have the signing or verifying complexity directly proportional to the group size or the number of revoked members, or require updates of signing key or public key once revocation occurs. Recently, although an outstanding improvement has been proposed in [20], the size of public key in the scheme is directly proportional to the group size.

Very recently, the most efficient method of this kind is proposed in [21]. It has constant signing and verifying complexity, and constant size in signature, public key, and signing key. Also it does not require updates of public key or signing key when member joining or leaving occurs. Thus, once Handauth is built on the scheme of [21], it can achieve scalability and dynamic participation. The time of each protocol run is independent of the number of MUs and revoked users. More specifically, it is constant. Thus, Handauth is efficient even in a large-scale network with many subscribers and many revoked users. However, we notice that a direct application of the method in [21] is still unable to meet Requirements 2-4 and 6-12, which are very challenging for ensuring secure, efficient, and robust access services for mobile users. For example, due to the sole network layer protocol design, it cannot preserve user anonymity and untraceability. Also, it does not provide Handauth with easily scheduled revocation, local access service expiration, attack-resistance, and AAA server anonymity properties.

To address these issues, some additional mechanisms are incorporated into the design of Handauth. Detailed description of these mechanisms will be given in Sections 4 and 5.

3.2 Overview of FSR-GS Technique

In this section, we first review the definition of FSR-GS.

Definition. A FSR-GS [21] is a tuple (G.Kg, G.Enroll, G.Revoke, G.Sign, G.Ver, G.Open) of probabilistic polynomial-time algorithms and one interactive mechanism. The parties involved in the FSR-GS include a group manager, a group member (i.e., a signer) and a verifier.

1. $G.Kg$. The group manager runs this algorithm to generate a master public key mpk, a master secret key msk (for enrolling group members), a trace key tk (for opening a signature), and an initial membership information Ω.
2. $G.Enroll$. This is an interactive procedure running between the group manager and a new user. Through this procedure, the new member U_i obtains a user signing key usk_i, a (public) user membership key upk_i, and a user revocation key rvk_i.
3. $G.Revoke$. On input mpk, rvk_i (of member U_i) and the current membership information Ω, the group manager outputs an updated Ω.
4. $G.Sign$. It takes mpk, upk_i, usk_i, rvk_i, Ω and a message m, and outputs a group signature σ.
5. $G.Ver$. On input mpk, Ω, m, and σ, it outputs 1 or 0 indicating acceptance or rejection on the validity of the signature σ on message m.
6. $G.Open$. On input mpk, tk, Ω and a valid message-signature pair (m, σ), the group manager outputs the user membership key upk_i of the actual signer.

Next, we review a concrete FSR-GS in [21], which will be employed in Handauth. Note that any other efficient forward secure revocable group signature schemes can just as easily be applied in Handauth.

Master-key generation$(G.Kg)$. The group manager randomly picks security parameters $\epsilon > 1, k, l_p \in N$ and then chooses the following parameters $\lambda_1, \lambda_2, \gamma_1$, and γ_2 such as $\lambda_1 > \epsilon(\lambda_2 + k) + 2, \lambda_2 > 4l_p, \gamma_1 > \epsilon(\gamma_2 + k) + 2$, and $\gamma_2 > \lambda_1 + 2$. All these parameters are public. $mpk = (n, a, a_0, y, g, h, g_1, g_2)$ and $msk = (p', q', x)$ are computed as follows: 1) Select random l_p-bit primes p', $q^{'}$ such that $p = 2p' + 1$ and $q = 2q' + 1$ are prime (Lagrange has proved this theorem: Let $p' \equiv 3(mod\,4)$ be prime. $2p' + 1$ is also prime if and only if $2p' + 1$ divides a mersenne prime number). Set the modulus $n = pq$. Note that all the arithmetic operations in the following sections are modulo n unless specified otherwise. 2) Choose random elements $a, a_0, g, h, g_1, g_2 \in_R QR(n)$ (of order $p^{'}q^{'}$). Here, $QR(n)$ presents the set of quadratic residues of group $\mathbb{Z}_n^*$. 3) Choose a random secret $x \in_R \mathbb{Z}^*_{p^{'}q^{'}}$ and $y = g^x$. The membership information is $\Omega = (c, \mu)$, where c is initialized to g_1 and μ is initialized to 1. Define the integral range $\Gamma = [2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}]$.

Enrollment$(G.Enroll)$. The upk_i, rvk_i, usk_i of the new member U_i are generated as follows. U_i randomly chooses $\widetilde{x}_i \in_R [0, 2^{\lambda_2}]$ and $\widetilde{r}_i \in_R [0, n]$ and then sends $C_1 = g^{\widetilde{x}_i} h^{\widetilde{r}_i}$ to the group manager. If $C_1 \in QR(n)$, the group manager randomly picks $\alpha_i, \beta_i \in_R [0, 2^{\lambda_2}]$ and sends (α_i, β_i) to member U_i. Member U_i generates $x_i = 2^{\lambda_1} + (\alpha_i \widetilde{x}_i + \beta_i \bmod 2^{\lambda_2})$ in $\mathbb{Z}$ and sends the group manager the value $C_2 = a^{x_i}$. Then, the group manager checks that $C_2 \in QR(n)$. If this is the case and all the proofs (detailed information can refer to [21]) were correct, the group manager picks a random prime $e_i \in_R \Gamma$ and generates $A_i = (C_2 a_0)^{1/e_i} = (a^{x_i} a_0)^{1/e_i}$. Then, the group manager sets $upk_i = A_i, rvk_i = e_i$. After that, the group manager sends $\{A_i, e_i\}$ to member U_i. Subsequently, member U_i verifies that $a^{x_i} a_0 = {A_i}^{e_i}$. If this is the case, member U_i sets $upk_i = A_i$, $rvk_i = e_i$ and $usk_i = x_i$.

Member revocation$(G.Revoke)$. On input of rvk_k of member U_k who is to be deleted at this time and the current $\Omega = (c, \mu)$, the group manager updates c as $c = c^{rvk_k}$ and updates μ as $\mu = \mu \cdot rvk_k$. Suppose there are currently revoked members $U_j, \ldots, U_k$, the latest c and μ become

$$c = g_1^{\prod_{i=j}^{k} rvk_i}$$

and $\mu = \prod_{i=j}^{k} rvk_i$.

Group signature generation$(G.Sign)$. A group signature σ on message m consists of a tuple $\widetilde{V_1}$, $\widetilde{V_2}$, where $\widetilde{V_1}$ and $\widetilde{V_2}$ are signatures of knowledge. The detailed description about these is given in [21].

Group signature verification$(G.Ver)$. To verify a group signature $\sigma = (\widetilde{V_1}, \widetilde{V_2})$ on message m and the revocation membership information Ω (actually only c is required), the verifier is to check the validation and correctness of $\widetilde{V_1}$, $\widetilde{V_2}$ with respect to mpk and Ω. The detailed information is given in [21].

Member trace$(G.Open)$. Given a message-signature pair $(m, \sigma = (\widetilde{V_1}, \widetilde{V_2}))$ and the trace key $tk = x$, if G.Ver$(mpk, tk, \Omega, m, \sigma) = 1$ then output the upk_i which is computed as $upk_i = T_1/T_2^x$.

4 Handauth: The Protocol

4.1 System Setup Phase

In Handauth, we have the following system setup:

1. The AAA server acts as the group manager of an FSR-GS system and has a master key pair (mpk, msk) and the initial membership information $\Omega = (c, \mu)$ generated using $G.Kg$, where $c = g_1$ and $\mu = 1$. Additionally, the AAA server also has a signing/verification key pair (sk, pk) of a conventional digital signature scheme, e.g., Elliptic Curve Digital Signature Algorithm (ECDSA).
2. The AAA server issues the master public key mpk to all APs. Additionally, each AP shares a session key AK_{AP} with the AAA server, respectively. As we discuss later, such a session key can be used to achieve Requirement 6.
3. The entire service provision time is divided into time intervals in the unit of hour, day, or month. We assume the AAA server sets day as the interval unit. In this case, the time interval has the format "YYYY/MM/DD." At the beginning of each day, each AP downloads the latest membership information Ω from the AAA server.
4. Each AP has a signing/verification key pair (sk_{AP}, pk_{AP}) of a conventional digital signature scheme, e.g., ECDSA. The ID and pk_{AP} of each AP are publicly known to all the users who are within the network controlled by the AP. This could be realized by requiring the visited AP to broadcast its digital certificate as part of *beacon messages* that are periodically broadcasted to declare service existence. In order that each MU is able to use the verification key pk_{AP} of the AAA server to verify that the serving AP is authorized by the AAA server to offer access services (i.e., Requirement 9), the digital certificate should be issued by the AAA server. Alternatively, when subscriber U_i registers to the AAA server, the certificates of all APs are loaded on U_i (e.g., built in the web browsers of all subscribers). The visited AP also broadcasts the latest membership information $\Omega = (c, \mu)$. Suppose for an AAA server, there are currently revoked subscribers $U_j, \ldots, U_k$, the latest

$$c = g_1^{\prod_{i=j}^{k} rvk_i}$$

and $\mu = \prod_{i=j}^{k} rvk_i$. Since user revocation key rvk_i of each user U_i is secretly shared between U_i and the AAA server, no one except the AAA server can learn any information from the membership information Ω. Each MU can verify those two information by using the AAA server's public key pk.

4.2 New User Joining Phase

Before accessing the network, an MU has to authenticate itself to the AAA server by in-person contact. For subscriber U_i, the AAA server runs $G.Enroll$ to generate a user signing key usk_i, a (public) user membership key upk_i, and a user revocation key rvk_i. The AAA server delivers all these keys and pk to U_i using a secure transmission protocol (e.g., wired transport layer security protocol). Note that the AAA server maintains a subscriber list, which is composed of every subscriber's related keys (e.g., user membership key, user revocation key) and expiration time. It is clear that different subscribers may have different expiration time. Obviously, the above procedure is invoked whenever a user wants to register with the AAA server.

4.3 Handover Authentication Phase

The handover authentication protocol which is carried out between a mobile user U_i and the visited access point $AP2$ is as follows. U_i first sends a login request to $AP2$ for mutual authentication. Then, $AP2$ checks the validity of U_i, establishes a session key and then gives a response to U_i. Subsequently, U_i validates $AP2$, establishes the session key and then responds to $AP2$. Finally, $AP2$ notifies the AAA server of the authentication result. We illustrate this procedure in Fig. 2, and the detailed steps are described as follows:

1. U_i first chooses a random number R_u, and a temporary identity $alias$ (not correlated in any way with the true user identity), and generates $\sigma_i = \text{G.Sign}(mpk, upk_i, usk_i, rvk_i, \Omega, alias \| g^{R_u} \| ts)$, where a time stamp ts is added by U_i to

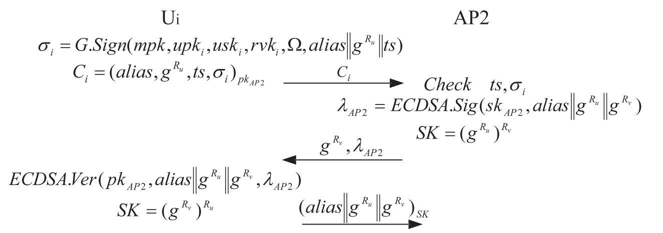
Ui
AP2
$\sigma_i = G.Sign(mpk, upk_i, usk_i, rvk_i, \Omega, alias\|g^{R_u}\|ts)$
$C_i = (alias, g^{R_u}, ts, \sigma_i)_{pk_{AP2}}$
C_i
Check ts, σ_i
$\lambda_{AP2} = ECDSA.Sig(sk_{AP2}, alias\|g^{R_u}\|g^{R_v})$
$SK = (g^{R_u})^{R_v}$
g^{R_v}, λ_{AP2}
$ECDSA.Ver(pk_{AP2}, alias\|g^{R_u}\|g^{R_v}, \lambda_{AP2})$
$SK = (g^{R_v})^{R_u}$
$(alias\|g^{R_u}\|g^{R_v})_{SK}$

Fig. 2. Authentication procedure of Handauth.

counter replay attacks. Subsequently, to meet Requirement 7, U_i encrypts the message $\{alias, g^{R_u}, ts, \sigma_i\}$ using $AP2$'s public key pk_{AP2} to produce $C_i = (alias, g^{R_u}, ts, \sigma_i)_{pk_{AP2}}$. Here, we write $(X)_K$ for message X encrypted with key K. Then, U_i sends the login request C_i to $AP2$.

2. After receiving C_i, $AP2$ uses its private key sk_{AP2} to decrypt it and then obtains the secret information $\{alias, g^{R_u}, ts, \sigma_i\}$. Subsequently, $AP2$ first checks whether the time stamp ts is within some allowable range compared with its current time. If it is positive, $AP2$ runs $G.Ver$ to verify whether the group signature σ_i is valid or not. If it is not valid, $AP2$ rejects the login request; otherwise, $AP2$ chooses a random number R_v, and computes $\lambda_{AP2} = ECDSA.Sig(sk_{AP2}, m_{AP2})$, where $m_{AP2} = alias\|g^{R_u}\|g^{R_v}$. Then, $AP2$ sends $\{g^{R_v}, \lambda_{AP2}\}$ back to U_i. Subsequently, $AP2$ computes the session key $SK = (g^{R_u})^{R_v}$ and erases R_v from its memory.
3. Upon receiving $\{g^{R_v}, \lambda_{AP2}\}$, U_i verifies λ_{AP2} by running ECDSA.Ver$(pk_{AP2}, m_{AP2}, \lambda_{AP2})$. If ECDSA.Ver returns 1, U_i generates the session key $SK = (g^{R_v})^{R_u}$ and erases R_u from its memory. After that, U_i generates $(alias\|g^{R_u}\|g^{R_v})_{SK}$ through symmetric encryption and then sends it to $AP2$. After receiving the message, $AP2$ decrypts and then verifies it. If the message is valid, $AP2$ concludes that U_i has established a session key and proceeds to the next step; otherwise, $AP2$ rejects the connection.
4. Finally, $AP2$ uses the secret key AK_{AP2} to encrypt the group signature message $\{alias, g^{R_u}, ts, \sigma_i\}$ and then delivers it to the AAA server. Upon receiving this message, the AAA server can obtain the identity of U_i by computing *G.Open*, which means that the AAA server can provide conditional privacy. Thus, it is shown that Handauth can achieve Requirement 6. Since APs only notify the AAA server of the authentication result after performing the handover authentication, this step does not affect the authentication time.

5 Discussion

5.1 Supporting Local Access Service Expiration

As described in Section 4.2, the AAA server maintains a subscriber list, which is composed of every subscriber's related keys (e.g., user membership key, user revocation key) and expiration time. Once a subscriber U_i's service subscription expires, its signing key usk_i should be invalidated from then on. In this case, the AAA server needs to run $G.Revoke(mpk, rvk_i, \Omega)$. That is, on input of rvk_i of member U_i and the current $\Omega = (c, \mu)$, the AAA server updates c as $c = c^{rvk_i}$ and updates μ as $\mu = \mu \cdot rvk_i$. This shows that Handauth can achieve Requirement 8.

5.2 Supporting Scheduled Revocation Easily

In practice, some users may need a predefined revocation period. For example, a mobile phone user may want to suspend the services for three months. A natural method is for such a user to reregister to the AAA server and then receive a new user signing key, a new user membership key, and a new user revocation key. Obviously, this method causes inconveniences. Therefore, to address this issue, Handauth provides a feasible approach as follows. We assume that a subscriber U_m is revoked at the interval t_1 and hopes to resume the services of the same AAA server with his previous keys (i.e., user signing key, user membership key, and user revocation key) at the interval t_2, where $t_2 > t_1$. At t_1, on input of rvk_m of subscriber U_m and the current $\Omega = (c, \mu)$, the AAA server updates c as $c = c^{rvk_m}$ and updates μ as $\mu = \mu \cdot rvk_m$. At t_2, on input of rvk_m of subscriber U_m and the current

$$\Omega = (c, \mu) = \left(g_1^{\prod_{i=j}^{k} rvk_i}, \prod_{i=j}^{k} rvk_i \right),$$

the AAA server updates c as $c = g_1^{\mu / rvk_m} = g_1^{\prod_{i=j}^{k} rvk_i / rvk_m}$ and then updates μ as $\mu = \mu / rvk_m$, where $U_m \in \{U_j, \ldots, U_k\}$. Subsequently, U_m resumes the services automatically and exactly at t_2, without the necessity to visit the AAA server. Hence, Handauth can satisfy Requirement 10.

5.3 Provision of Dynamic User Revocation Mechanism

There may be misbehaving users in the system. In this case, the AAA server can identify these misbehaving users in step 4 of the handover authentication procedure, and then revoke them through running $G.Revoke(mpk, rvk_i, \Omega)$. Therefore, Handauth can meet Requirement 11.

5.4 Cross-Layer Protocol Design for Strong User Anonymity and Untraceablility

In the MAC layer, the restriction on handover authentication with user anonymity is that the standards of current wireless technologies, such as IEEE 802.11 and Bluetooth, require manufacturers to assign an identification number (i.e., MAC address) to every device (i.e., Laptop PC). The MAC address is like an annoying tag attached to a mobile device, anytime, and anywhere. Obviously, such a practice exposes the ID of a mobile device at the MAC address. However, current handover authentication techniques [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] do not consider this security issue. For Handauth, an ideal way to remedy this weakness is to replace the MAC address with a user alias. Alias collision should not be a serious problem in this case and can be prevented in many ways, for instance, by adding a time stamp or random number.

In the physical layer, various noncryptographic techniques for user authentication and device identification in wireless networks using physical layer properties or information (e.g., frame sequence number, packet size, and signal strength) have been suggested [22]. This fact gives the adversary the techniques of tracking the targeted MUs. For example, the packet sizes of the MUs have been exploited to identify different users [23]. Since all existing handover authentication solutions [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] do not consider this issue, they fail to provide user anonymity and untraceability. Obviously, in order to resist these attacks, some countermeasure to these techniques should be employed in mobile user side of Handauth. A feasible way is that each MU should frequently change its physical layer properties or information. For instance, to address this issue, each MU should frequently change its frame sequence number, packet size, and signal strength, by using some ways (e.g., random number generator).

6 Security Analysis

6.1 Simple Proof of Security Requirements

We analyze the security of Handauth with respect to the security requirements given in Section 1. As described in Sections 4 and 5, it is clear that Handauth can meet Requirements 2, 5, 6, 8, 10, and 11. Due to the use of forward secure revocable group signature,

TABLE 1
Time Costs for Cryptography Operations

(ms)	T_{PM}	T_{MI}	T_{ME}	T_{RV}	T_{TP}
MU	1.082	0.653	1.019	0.435	33.584
AP	0.376	0.334	0.387	0.201	11.903

T_{PM}: point multiplication, T_{MI}: modular inverse, T_{ME}: modular exponentiation, T_{RV}: RSA verification, T_{TP}: tate pairing

Handauth can achieve Requirement 4. The reason would be clear when readers refer to the forward secure user revocation definition for FSR-GS in [21]. In the following, we focus on how Handauth meets Requirements 1, 7, 9, and 12.

Mutual authentication. AP authentication is done by the challenge-response pair $(g^{R_u}, ECDSA.Sig(sk_{AP2}, alias\|g^{R_u}\|g^{R_v}))$, by which a user is sure about the identity of the visited AP. Since only $AP2$ has sk_{AP2}, no other APs can compute a valid digital signature on U_i's freshly generated challenge g^{R_u}. It should be noted that only the AAA server can generate a valid certificate for $AP2$, and the identity of $AP2$ and its public key pk_{AP2} are included and bound by the certificate. Therefore, other APs cannot cheat by using different public keys or different IDs. Thus, Handauth can satisfy Requirement 9. Since the group signature message $\{alias, g^{R_u}, ts, \sigma_i\}$ is encrypted using $AP2$'s public key pk_{AP2}, only $AP2$ can use its private key sk_{AP2} to obtain such a group signature message and then obtain the identity of U_i's AAA server. Thus, the identity of U_i's AAA server can be hidden from eavesdroppers and the legitimate network entities except the visited AP (i.e., $AP2$). That is, Handauth can meet Requirement 7. Subscriber authentication is achieved by another challenge-response pair: $(g^{R_u}, ts,$ G.Sign $(mpk, upk_i, usk_i, rvk_i, \Omega, alias\|g^{R_u}\|ts))$. Only a legitimate subscriber of the AAA server can generate a valid group signature on U_i's challenge $\{g^{R_u}, ts\}$ and the current member message Ω. Thus, Handauth can satisfy Requirement 1. According to the above analysis, Handauth can provide mutual authentication.

Strong user anonymity and untraceablility. User anonymity is achieved by the anonymity of G.Sign $(mpk, upk_i, usk_i, rvk_i, \Omega, alias\|g^{R_u}\|ts)$. An adversary (including eavesdroppers) and APs are not able to obtain the identity of the real signer since they do not have the trace key tk, which is preserved only by the AAA server. That is to say, when the handover authentication runs, the visited AP is just able to determine whether an MU is the subscriber of the AAA server, but it cannot derive any further identity information about the MU. User untraceability is also achieved by the anonymity of the group signature. The reason would become clear when readers refer to the anonymity definition and security analysis for FSR-GS in [21].

6.2 Formal Analysis Using AVISPA

Besides the above analysis, we also provide a formal analysis of Handauth here. Many formal security validation approaches and tools have been proposed in the literature. In this paper, we choose AVISPA [24], for the following reasons: First, it is expressive enough and we can model several properties like secrecy of keys, authentication, freshness, robustness against replay attacks, etc. Second, it provides a user friendly specification language called the High-Level Protocol Specification Language (HLPSL) [25] for specifying targeted protocols and formally validating them. Third, it is widely used by developers of security protocols and by academic researchers to analyze possible attacks on security protocols [16].

TABLE 3
Comparison of Communication Overhead

[5]	[7], [14], [15]	[6]	[13]	[8]	[9]	[10]- [12]&Handauth
2+5δ	2+2δ	4+4δ	3+2δ	4δ+1ϱ	3δ+2ϱ	3δ

The current version of the AVISPA tool integrates the following four back ends: On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). The attacker implemented in AVISPA is a Dolev-Yao attacker [26], which can overhear, intercept messages, inject new messages or modify messages in transit. Therefore, AVISPA is appropriate for the analysis of security protocols in wireless networks. Our proposed protocol has been translated to HLSPL. The HLPSL specification uses participants to enact each role, and also specifies how many concurrent sessions of the protocol are running. Overall, three sessions of the protocol were modeled and checked concurrently, to ensure that the goal is realized.

Once the HLPSL specification has been debugged, it was checked automatically for attack detection using the AVISPA verification tools. No revealed attacks were found, and the security goals are achieved. The whole test results are given as follows:

1. OFMC reports the mechanism is safe.
2. CL-AtSe reports the mechanism is safe.
3. SATMC reports the mechanism is safe.
4. TA4SPS reports that some rules may be not fired, so it does not do the verification.

Therefore, the AVISPA cannot produce any attack on Handauth.

7 Performance and Implementation

We consider two performance metrics: authentication latency and communication overhead. We investigated the time costs of the primitive cryptography operations using the OpenSSL library [27] on an Intel P III Mobile 733 MHz processor as an MU and Intel P IV 3 GHz as an AP in Table 1. In the experiment, the key sizes of ECC and RSA are set to 160 bits and 1,024 bits, respectively. In this paper, the authentication latency is defined as the time of cryptography operations. Note that the time costs of highly efficient operations such as hash function, symmetric encryption/decryption, and point addition are omitted.

Table 2 compares the authentication delay of Handauth and related works ([10], [11]). Here, T_{MU} and T_{AP} denote the authentication latency on an MU and an AP, respectively. Also, $T_{MU_{opt}}$ indicates the authentication time optimized by precomputation on an MU. From Table 2, it can be seen that a successful handover authentication in Handauth requires eight modular exponentiations, 3.25 point multiplications, and 1 modular inverse computation (plus 13 modular exponentiations that can be precomputed) on an MU. And it requires 18 modular exponentiations and 1 modular inverse computation on an AP without other precomputation requirements. Totally, a successful handover authentication in Handauth requires 19.622 ms. Currently, the clock frequencies of most Laptop PCs, PDAs and smart phones are

TABLE 2
Comparison of Authentication Delay

	SFRIC [10]	method of [11]	Handauth
T_{MU}	T_{PM}+2T_{TP}=68.25ms	4T_{ME}+T_{RV}=4.5ms	8T_{ME}+3.25T_{PM}+$T_{MI} = 12.3ms$
$T_{handover} = T_{MU\_opt}$+$T_{AP\_opt}$	T_{TP}+2T_{TP}=57.4ms	2×(3T_{ME}+1T_{RV})=4.9ms	(8T_{ME}+3.25T_{PM}+T_{MI})+(18T_{ME}+T_{MI})=19.6ms

TABLE 4
Functionality Comparison between Handauth and Related Work

Protocols	HMZCJ [14]	Scheme of [13]	CHCBGR [16]	SFRIC [10]	method of [11]	CJKY [12]	Handauth
Number of parties	3	3	3	2	2	2	2
Strong user anonymity and untraceablility	No	No	No	No	No	No	Yes
Conditional privacy preservation	Yes	Yes	Yes	No	No	No	Yes
AAA server anonymity	No	No	No	No	No	No	Yes
Local access service expiration	No	No	No	Yes	Yes	Yes	Yes
Local AP validation	No	No	No	No	No	No	Yes
Easily scheduled revocation	No	No	No	No	No	No	Yes
Dynamic user revocation	Yes	Yes	Yes	No	No	No	Yes
Attack-resistance	No	No	No	No	No	No	Yes

greater than 700 MHz. Therefore, Handauth is efficient to be employed on most mobile devices.

Table 3 compares the communication overhead of Handauth and related works [5], [6], [7], [8], [9], [10], [11], [12], [14], [15]. For communication overhead, we assume that the expected authentication message delivery cost between an AP and the AAA server is one unit and that between an MU and an AP is δ unit, respectively. The cost δ unit is within the range $0 < \delta < 1$ since the AAA server is often located in a remote location. The cost ϱ between the APs is generally lower than one unit. As shown in Tables 2 and 3, compared to related works [5], [6], [7], [8], [9], [10], [11], [12], [14], Handauth enjoys both computation and communication efficiency.

Finally, we make the functionality comparisons of Handauth and the well-known approaches [10], [11], [12], [13], [14], [16] in Table 4. It can be seen that only our scheme achieves all security requirements.

8 Conclusion

In this paper, we have identified the characteristics of access services for mobile users and concluded 12 properties that an efficient handover authentication scheme should satisfy. Moreover, we have proposed a novel protocol named Handauth to achieve secure and efficient handover authentication. The protocol satisfies a set of important requirements which have not been addressed by earlier works. The security analysis and experimental results show that the proposed approach is feasible for real applications. Further, the security properties of Handauth have been formally verified.

References

[1] European Telecomm. Standards Inst. (ETSI), GSM 02.09: Security Aspects, 1993.

[2] 3rd Generation Partnership Project, 3GPP Specification: 3GPP TS 33.102, 3G Security, Security Architecture, Dec. 2002.

[3] P. Funk and S. Blake-Wilson, “EAP Tunneled TLS Authentication Protocol (EAP-TTLS),” *IETF Internet Draft,* draft-ietf-pppext-eap-ttls-05.txt, July 2004.

[4] A. Palekar et al., “Protected EAP Protocol (PEAP),” *IETF Internet Draft,* draft-josefsson-pppext-eap-tls-eap-06.txt, Mar. 2003.

[5] S. Pack and Y. Choi, “Fast Handoff Scheme Based on Mobility Prediction in Public Wireless LAN Systems,” *Proc. IEE Comm.,* vol. 151, no. 5, pp. 489-495, Oct. 2004.

[6] C.-C. Chang, C.-Y. Lee, and Y.-C. Chiu, “Enhanced Authentication Scheme with Anonymity for Roaming Service in Global Mobility Networks,” *Computer Comm.,* vol. 32, no. 4, pp. 611-618, 2009.

[7] D. He and S. Chan, “Design and Validation of an Efficient Authentication Scheme with Anonymity for Roaming Service in Global Mobility Networks,” *Wireless Personal Comm.,* vol. 61, no. 2, pp. 465-476, 2011.

[8] H. Wang and A.R. Prasad, “Fast Authentication for Inter-domain Handover,” *Proc. Int'l Conf. Telecomm. (ICT '04),* 2004.

[9] J. Choi and S. Jung, “A Secure and Efficient Handover Authentication Based on Light-Weight Diffie-Hellman on Mobile Node in FMIPv6,” *IEICE Trans. Comm.,* vol. E-91B, no. 2, pp. 605-608, 2008.

[10] Y. Kim, W. Ren, J. Jo, M. Yang, Y. Jiang, and J. Zheng, “SFRIC: A Secure Fast Roaming Scheme in Wireless LAN Using ID-Based Cryptography,” *Proc. Int'l Conf. Comm. (ICC '07),* 2007.

[11] J. Choi and S. Jung, “A Handover Authentication Using Credentials Based on Chameleon Hashing,” *IEEE Comm. Letters,* vol. 14, no. 1, pp. 54-56, Jan. 2010.

[12] J. Choi, S. Jung, Y. Kim, and M. Yoo, “A Fast and Efficient Handover Authentication Achieving Conditional Privacy in V2I Networks,” *Proc. Int'l Conf. Smart Spaces and Next Generation Wired/Wireless Networking,* S. Balandin et al., eds., pp. 291-300, 2009.

[13] D. He and S. Chan, “A Secure and Lightweight User Authentication Scheme with Anonymity for the Global Mobility Network,” *Proc. Int'l Conf. Network-Based Information Systems (NBiS '10),* pp. 305-312, 2010.

[14] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, “A Strong User Authentication Scheme with Smart Cards for Wireless Communications,” *Computer Comm.,* vol. 34, no. 3, pp. 367-374, 2011.

[15] C.-C. Chang and H.-C. Tsai, “An Anonymous and Self-Verified Mobile Authentication with Authenticated Key Agreement for Large-Scale Wireless Networks,” *IEEE Trans. Wireless Comm.,* vol. 9, no. 11, pp. 3346-3353, Nov. 2010.

[16] C. Chen, D. He, S. Chan, J. Bu, Y. Gao, and R. Fan, “Lightweight and Provably Secure User Authentication with Anonymity for the Global Mobility Network,” *Int'l J. Comm. Systems,* vol. 24, no. 3, pp. 347-362, 2011.

[17] D. Samfat, R. Molva, N. Asokan, “Untraceability in Mobile Networks,” *Proc. MobiCom '95,* pp. 26-36, 1995.

[18] E. Bresson and J. Stern, “Efficient Revocation in Group Signatures,” *Proc. Conf. Public Key Cryptography (PKC '01),* 2001.

[19] T. Nakanishi and Y. Sugiyama, “A Group Signature Scheme with Efficient Membership Revocation for Reasonable Groups,” *Proc. Australasian Conf. Information Security and Privacy (ACISP '04),* 2004.

[20] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki, “Revocable Group Signature Schemes with Constant Costs for Signing and Verifying,” *Proc. Conf. Public Key Cryptography (PKC '09),* 2009.

[21] H. Jin, D. Wong, and Y. Xu, “Efficient Group Signature with Forward Secure Revocation,” *Proc. Int'l Conf. Security Technology (SecTech '09),* 2009.

[22] K. Zeng, K. Govindan, and P. Mohapatra, “Non-Cryptographic Authentication and Identification in Wireless Networks,” *IEEE Wireless Comm.,* vol. 17, no. 5, pp. 56-62, Oct. 2010.

[23] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, “802.11 User Fingerprinting,” *Proc. MobiCom '07,* pp. 99-110, 2007.

[24] E.M. Clarke et al., *Model Checking.* MIT Press, 1999.

[25] J. Daemen and V. Rijmen, *The Design of Rijndael.* Springer-Verlag, 2002.

[26] D. Dolev and A. Yao, “On the Security of Public Key Protocols,” *IEEE Trans. Information Theory,* vol. 29, no. 2, pp. 198-208, Mar. 1983.

[27] OpenSSL, http://www.openssl.org, 2012.