Distributed Anonymous Authentication in Heterogeneous Networks

Shin-Ming Cheng and Cheng-Han Ho
Department of Computer Science
and Information Engineering,
National Taiwan University
of Science and Technology,
Taipei 106, Taiwan.
{smcheng, m10115061}@mail.ntust.edu.tw

Shannon Chen
Department of Computer Science,
University of Illinois at Urbana-Champaign
Illinois, USA
cchen116@illinois.edu

Shih-Hao Chang
Department of Computer Science
and Information Engineering,
Tamkang University
New Taipei City, Taiwan
sh.chang@ieee.org

***Abstract*—Nowadays, the design of a secure access authentication protocol in heterogeneous networks achieving seamless roaming across radio access technologies for mobile users (MUs) is a major technical challenge. This paper proposes a Distributed Anonymous Authentication (DAA) protocol to resolve the problems of heavy signaling overheads and long signaling delay when authentication is executed in a centralized manner. By applying MUs and point of attachments (PoAs) as group members, the adopted group signature algorithms provide identity verification directly without sharing secrets in advance, which significantly reduces signaling overheads. Moreover, MUs sign messages on behalf of the group, so that anonymity and unlinkability against PoAs are provided and thus privacy is preserved. Performance analysis confirms the advantages of DAA over existing solutions.**

Keywords-anonymous authentication, group signature, heterogeneous networks

I. Introduction

The next generation networks are expected to exhibit heterogeneity of Radio Access Technologies (RATs) consisting of various wireless networks (e.g., WiMAX and WiFi) and cellular communications (e.g., WCDMA and HSPA). These heterogeneous RATs are integrated together to provide mobile users (MUs) the universal wireless access. However, the convenience of user mobility increases the risk of being masqueraded and eavesdropped by malicious users. Thus, the access authenticity and user privacy when MUs connect to various RATs should be ensured.

Authentication is used as an initial process to authorize an MU for communication by an authenticator through secret credentials negotiation and challenge/response verification. The data confidentiality and integrity over an RAT can be further protected by the security association build on the negotiated secret credentials. Except the eavesdroppers, an MU is desirable to keep anonymous from foreign domain when it changes serving RAT, and only home domain can recover the MU's real identity [1]. This prevents MU's moving history and current location from being tracked, thus the anonymity and intractability of user identity are achieved.

Traditionally, the MU authentication relies on a costly, centralized authenticator, which assigns the subscribed MU a corresponding secret credential. Each time the MU handovers across administrative domains or switches serving RAT, the authentication procedure involving the authenticator in home domain causes long signaling delay and the dissatisfaction of QoS for delay-sensitive and real-time services may not be avoided [2]. Moreover, centralized authenticator maintains user anonymity by associating an *alias* to the MU's real identity in the whole authentication procedure [1] and thus user unlinkability can be achieve [3], where foreign domain cannot link the two different connections initiated by the same unknown MU to build any user profile. However, the verification for the roaming MU in foreign domains generates extra signal overhead and delay due to it's centralized architecture. Moreover, large synchronization cost occurs due to *alias* has to be renewed every time after being used.

To tackle the above challenges, various kinds of distributed authentication solutions are proposed. The most famous one adopts certificate as a reliable combination of user identification and public key, which is issued by a trusted certificate authority (CA) [4]. Various schemes like public-key-based authentication [5], and password-based authentication [6] are proposed; however, some common drawbacks exist. Each time before an authentication begins, the exchange of certificates between the MU and the point of attachment (PoA) introduces extra communication overhead. Moreover, the verification of certificate of the PoA massively increases the computational loads. In addition, the protocols are not sophisticatedly designed to support various mobility scenarios (like inter-domain and intra-domain handovers), which incurs communication and computational overheads.

Identity-based cryptography (IBC) [7], [8] applied in authentication schemes [9] utilizes an information that uniquely identifies users rather then a user public key to sign data, thus central PKI becomes unnecessary. While this alleviates communication loads from MU, user anonymity and unlinkability against foreign domain are obviously sacrificed. Token-based authentication scheme [10] relies on PoAs to collectively store authentication information, which eliminates the need to maintain costly infrastructure required by the traditional centralized scheme. A token that promises the validity of MU are passed form the old PoA to the new one in handover scenarios, which reduces overhead of re-authentication. However, security issues like non-repudiation, fake PoA attack, and in-time revocation are not concerned.

While the authentication schemes mentioned above facing tradeoffs between security and efficiency, this paper proposes a protocol that upholds all necessary security criteria under minimum communication loads via distributed authentication at the expense of slight increase in computation complexity. With the advancing computational capability and increasing storage, mobile equipments become more powerful to undertake more complex operations. The bottleneck of efficiency tends to become the number of messages exchanged and the number of parties participated in the security procedures.

Under group signature-based authentication [11], [12], nodes verify each other as valid members belong to the claimed group rather then a specific identity. Therefore, the authentication between MU and it's current PoA can be accomplished directly without sharing secrets in advance, which reduces the number of messages exchanged and makes it more suitable for roaming scenarios in heterogeneous networks. In security aspects, the subjects of authentication are raised from nodes to groups in group signature schemes. Intruders have to compromise the whole group to proceed an undetectable attack. Moreover, any two group signatures generated by a node cannot be linked, thus the anonymous and unlinkable authentication is achieved.

The remainder of this paper is organized as follows: In section II, we survey the legacy group signature model. Section III describes the system model including network architecture and the proposed group signature model feasible to heterogeneous network. The proposed distributed anonymous authentication protocol is presented in IV. Section V gives the security and performance analysis of the proposed protocol, and finally, section VI gives the concluding remarks.

II. Background of Group Signatures

In group signature schemes, group members sign messages on behalf of the group, so that the anonymity of the signer is provided. However, the anonymity can be broken by group manager in case of legal disputes. In addition, in some group signature schemes, the group manager can dynamically alter the membership of the group by recruitment and revocation anytime after the group is established.

Bilinear pairing on elliptic curve recently received considerable attention (such as Weil pairing [8]) since the technique has been identified to be able to solve some problems that were previously well recognized as unsolvable. Another advantage in considering pairing-based schemes is that they can save communication bandwidth as compared with the traditional schemes, such as RSA and ElGamal, due to a smaller signature overhead. Typically, Bilinear pairing satisfies the following characteristics:

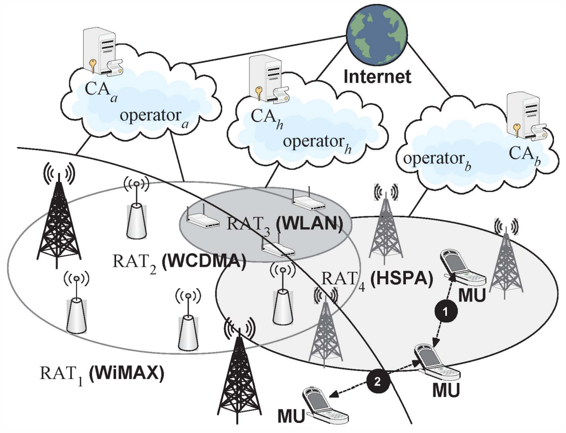

Fig. 1. Heterogeneous Network Architecture

Definition 1. *(Admissible Bilinear Map [8]): Let* $(\mathbb{G}_1, \times)$, $(\mathbb{G}_2, \times)$ *be two groups of order* q *for same large prime* q. *An admissible bilinear map is* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, $P, Q \in \mathbb{G}_1$ *and* $\forall a, b \in \mathbb{Z}_q^*$ *are satisfying the following properties.*

1) *Bilinearity:* $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$
   $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q)\hat{e}(P_2, Q)$
   $\hat{e}(P, Q_1 + Q_2) = \hat{e}(P, Q_1)\hat{e}(P, Q_2)$
2) *Nondegeneracy: There exists* P_1 *and* P_2 *be two generators of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *such that* $\hat{e}(P_1, P_2) \neq 1$*;in other words, the map does not send all pairs in* $G_1 \times G_2$ *to the identity in* G_T. $\hat{e}(P_1, P_2) \neq 1_{G_T}$.
3) *Computability: There exists an efficient algorithm to compute* $\hat{e}(P, Q)$, $\forall (P, Q) \in G_1 \times G_2$.

III. System Model

A. Network Architecture

In this paper, we consider a heterogeneous network architecture comprising a core network and multiple heterogeneous RATs, as shown in Fig. 1. Without loss of generality, this network consists of four different RATs, WiMAX, WCDMA, WLAN, and HSPA with indexes 1 to 4 respectively. We assume that RAT_1 and RAT_2 belong to $operator_a$, RAT_3 belong to $operator_h$, and RAT_4 belong to $operator_b$. An MU with ID ID_{MU} subscribed to domain h belongs to group G_h founded by CA_h. The CAs of domains are bridged together to communication with each other. Assume PoAs P_a and P_b located in domains a and b, respectively. Denote ID_{P_x} be the identity of PoA P_x, which is a member of group G_x founded by CA_x. Denote the group public keys of G_x as gpk_x. The member key pairs of the MU and the two PoAs are denoted as $\{msk_{MU}, stk_{MU}\}$, $\{msk_{P_a}, stk_{P_a}\}$, and $\{msk_{P_b}, stk_{P_b}\}$. In addition, CA_x is a member of its own group and its member key pairs are denoted by $\{msk_{CA_x}, stk_{CA_x}\}$. Denote RL_x as the revocation list of group G_x.

Three types of authentications for the MU are considered:

Session authentication: When an MU starts a communication session in a RAT, a session authentication is initiated.

Intra-domain handover authentication: When an MU is crossing the boundary of RATs in the same domain with an on-going service (e.g., movement 1 in Fig. 1), an intra-domain handover authentication is initiated.

Inter-domain handover authentication: When an MU is crossing the boundaries of different domains with an on-going service (e.g., movement 2 in Fig. 1), an inter-domain handover authentication occurs.

B. Proposed Group Signature Model

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two multiplicative cyclic groups with a generator g_1 and g_2 of the same prime order p, respectively. Let ψ be a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$. with $\psi(g_2) = g_1$, and $\hat{e}$ be a computable map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following properties. 1) Bilinearity: For all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, and a, $b \in \mathbb{Z}_p^*$, $\hat{e}(u, v)^{ab}$. 2) Nondegeneracy: $\hat{e}(g_1, g_2) = g \neq 1_{\mathbb{G}_T}$.

Furthermore, we assume that the strong DiffieHellma (SDH) assumption holds on $(\mathbb{G}_1, \mathbb{G}_2)$ and that the linear DiffieHellma assumption holds on $\mathbb{G}_1$ [13].

- $\mathsf{Est}(n)$ This randomized algorithm takes as input a parameter n, the number of members of the group, and proceeds as follows. Select a generator g_2 in $\mathbb{G}_2$ uniformly at random, and set $g_1 \leftarrow \psi(g_2)$. Select $h \overset{R}{\leftarrow} \mathbb{G}_1 \backslash 1_{\mathbb{G}_1}$ and $\varepsilon_1, \varepsilon_2 \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and set $u, v \in \mathbb{G}_1$ such that $u^{\varepsilon_1} = v^{\varepsilon_2} = h$. Select $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and set $w = g_2^{\gamma}$.Finally, $\mathsf{Est}(n)$ outputs $gmsk_1 = \gamma$ and $gpk = (g_1, g_2, h, u, v, w)$.The private key of the group manager is $gmsk_2 = (\varepsilon_1, \varepsilon_2)$.No party is allowed to possess γ; it is only known to the private-key issuer.
- $\mathsf{Rcr}(gpk, gmsk_1)$ Using γ generate for each user i, $1 \leq i \leq n$, an SDH tuple (A_i, x_i): select $x_i \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and set $A_i \overset{R}{\leftarrow} g_1^{1/(r+x_i)} \in \mathbb{G}_1$. Each users secret key is her tuple $gsk[i] = (A_i, x_i)$ and $stk = A_i$.
- $\mathsf{Sig}(gpk, gsk[i], M)$ Given a group public key $gpk = (g_1, g_2, u, v, w)$, a users key $gsk[i] = (A_i, x_i)$, and a message $M \in \{0, 1\}^*$, compute the signature as follows:
  1) Then it randomly selects $\alpha, \beta \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and Comput the value $T_1 \leftarrow u^{\alpha}$, $T_2 \leftarrow v^{\beta}$ and $T_3 \leftarrow A_i h^{\alpha+\beta}$. After setting $\delta_1 \leftarrow x_i\alpha$ and $\delta_2 \leftarrow x_i\beta \in \mathbb{Z}_p$ and selecting $r_{\alpha}, r_{\beta}, r_{x_i}, r_{\delta_1}$ and $r_{\delta_2} \overset{R}{\leftarrow} \mathbb{Z}_p^*$, it computes R_1, R_2, R_3, R_4 and R_5 respectively as $R_1 \leftarrow u^{r_{\alpha}}$, $R_2 \leftarrow v^{r_{\beta}}$, $R_3 \leftarrow \hat{e}(T_3, g_2)^{r_x} \times e(h, w)^{-r_{\alpha}-r_{\beta}} \times \hat{e}(h, g_2)^{-r_{\delta_1}-r_{\delta_2}}$, $R_4 \leftarrow T_1^{r_{x_i}} \times u^{-r_{\delta_1}}$ and $R_5 \leftarrow T_2^{r_{x_i}} \times v^{-r_{\delta_2}}$
  2) Compute a challenge c using the hash function as: $c \leftarrow H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in \mathbb{Z}_p^*$.
  3) Using c construct the values s_{α}, s_{β}, s_{x_i}, s_{δ_1} and s_{δ_2} are computed respectively as $s_{\alpha} \leftarrow r_{\alpha} + c\alpha$, $s_{\beta} \leftarrow r_{\beta} + c\beta$, $s_{x_i} \leftarrow r_{x_i} + cx_i$, $s_{\delta_1} \leftarrow r_{\delta_1} + c\delta_1$ and $s_{\delta_2} \leftarrow r_{\delta_2} + c\delta_2$
  4) Output the signature σ, computed as $\sigma = (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_{x_i}, s_{\delta_1}, s_{\delta_2})$.
- $\mathsf{Vrf}(gpk, M, \sigma)$ Given a group public key $gpk = (g_1, g_2, h, u, v, w)$, a message M, and a group signature σ , verify that σ is a valid signature as follows:
  1) Computes the $\tilde{R}_1$, $\tilde{R}_2$, $\tilde{R}_3$, $\tilde{R}_4$ and $\tilde{R}_5$ by $\tilde{R}_1 \leftarrow u^{s_{\alpha}} \times T_1^{-c}$, $\tilde{R}_2 \leftarrow v^{s_{\beta}} \times T_2^{-c}$, $\tilde{R}_3 \leftarrow e(T_3, g_2)^{s_{x_i}} \times \hat{e}(h, w)^{-s_{\alpha}-s_{\beta}} \times \hat{e}(h, g_2)^{-s_{\delta_1}-s_{\delta_2}} \times (\hat{e}(T_3, w)/e(g_1, g_2))^c$, $\tilde{R}_4 \leftarrow T_1^{s_{x_i}} \times u^{-s_{\delta_1}}$ and $\tilde{R}_5 \leftarrow T_2^{s_{x_i}} \times v^{-s_{\delta_2}}$, respectively.
  2) Using a value obtained by execution of the firs step, and the value of the signature σ, to check the $c \overset{?}{=} H(M, T_1, T_2, T_3, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4, \tilde{R}_5)$. Accepts if this check succeeds and reject otherwise.
- $\mathsf{Trc}(gpk, gmsk, M, \sigma)$ This algorithm is used for tracing a signature to a signer. It takes as input a group public key $gpk = (g_1, g_2, h, u, v, w)$ and the corresponding group managers secrt key $gmsk = (\varepsilon_1, \varepsilon_2)$, together with a message M and a signature $\sigma = (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_{x_i}, s_{\delta_1}, s_{\delta_2})$ to trace, and proceeds as follows. First, verify that σ is a valid signature on M. Second, consider the firs three elements (T_1, T_2, T_3) as a Linear encryption, and recover the users A_i as $A \leftarrow T_3/(T_1^{\varepsilon_1} \times T_2^{\varepsilon_2})$. If the group manager is given the elements $\{A_i\}$ of the users private keys, he can look up the user index corresponding to the identity A_i recovered from the signature.

IV. DAA Protocols

This section describes the system setup and the three authentication schemes of DAA protocol. By adopting the proposed group signature scheme, the following notations are used. As Fig. 1 shows, an MU with ID ID_{MU} subscribed to domain h belongs to group G_h founded by CA_h. Assume PoAs P_a and P_b located in domains a and b, respectively. Denote ID_{P_x} be the identity of PoA P_x, which is a member of group G_x founded by CA_x. Denote the group public keys of G_x as gpk_x. The member key pairs of the MU and the two PoAs are denoted as $\{gsk_{MU}, stk_{MU}\}$, $\{gsk_{P_a}, stk_{P_a}\}$, and $\{gsk_{P_b}, stk_{P_b}\}$. In addition, CA_x is a member of its own group and its member key pairs are denoted by $\{gsk_{CA_x}, stk_{CA_x}\}$. Denote RL_x as the revocation list of group G_x define in the previous section.

Every entity (including CA, PoA, and MU) possesses a cache called *keychain* for storing group public keys of other groups. Assume that CAs and PoAs have larger memory space reserved for the *keychain* then MUs does, and thus CAs and PoAs can possess the group public key of every group at any time while MUs can only store the group public keys that are more likely to be used again in the future in its *keychain*. Each group key is stored in the *keychain* of a MU in a 3-tuple manner: $\{ID_y, gpk_y, ts\}$, where ID_y is a group ID, gpk_y is the public key of that group, and ts is a time stamp.

A. Initialization Procedures

Step I1. A CA plays the role of a founder who establishes a group by activating $\mathsf{Est}()$. Let the generated group public key be gpk and the group secret key be gsk.

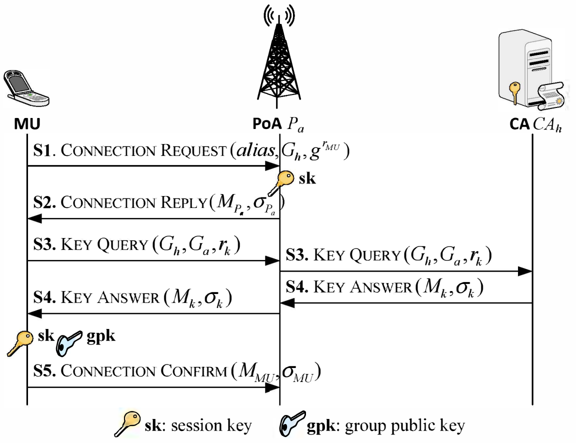
MU
PoA P_a
CA CA_h
S1. CONNECTION REQUEST $(alias, G_h, g^{r_{MU}})$
sk
S2. CONNECTION REPLY $(M_{P_\bullet}, \sigma_{P_a})$
S3. KEY QUERY (G_h, G_a, r_k)
S3. KEY QUERY (G_h, G_a, r_k)
S4. KEY ANSWER (M_k, σ_k)
S4. KEY ANSWER (M_k, σ_k)
sk
gpk
S5. CONNECTION CONFIRM (M_{MU}, σ_{MU})
sk: session key
gpk: group public key

Fig. 2. Message flow of the DAA protocol in session authentication

Step I2. After a group is established, the CA broadcasts its gpk to all the other CAs and request their gpks. By doing so, the public keys of all groups are publicly known by every CA in the scheme. In practice, this is done by utilizing a pre-established hard-wired network among CAs through bridge or centralized CA.

Step I3. The CA activates $\mathsf{Rcr}(gpk, gmsk)$ to recruit itself as a member of the group. In return, the CA gets its own $\{gpk, gsk\}$ pair.

Step I4. When the CA receives a new gpk from a new established CA, it adds the key into its own *keychain* then broadcasts it to all its PoA members. The PoA members receive the key and store it in their *keychain*s.

Step I5. When an owner of the PoA, may be the operator itself or otherwise, chooses a trusted CA to join its group, the CA assign an unique identity to it and activates $\mathsf{Rcr}(gpk, gmsk)$. In return, the PoA gets $\{ID, gpk, gsk\}$ and the CA records the $\{ID, stk\}$ pair of the PoA. In addition, the new PoA member is given the stk of the CA and all the gpks of other groups so that the PoAs have the gpks of every groups in their *keychain*.

Step I6. When an MU chooses a trusted CA to subscribe, the CA assigns an unique identity to it and activates $\mathsf{Rcr}(gpk, gmsk)$. In return, the MU gets $\{ID, gpk, gsk\}$ and the CA records the $\{ID, stk\}$ pair of the MU. In addition, the new MU member is given the stk of the CA.

B. Session Authentication Procedures

The message flows of DAA session authentication is illustrated in Fig. 2, and details are described as follows.

Step S1. When an MU starts a connection to a PoA (P_a), the MU randomly selects $r_{MU} \overset{R}{\leftarrow} \mathbb{Z}_p$ and an $alias$. Then the MU generates CONNECTION REQUEST $\leftarrow \{alias, G_h, g^{r_{MU}}\}$ and sends it to the PoA.

Step S2. After the PoA receives the request, it randomly selects $r_{P_\bullet}$ and computes a session key $K \leftarrow (g^{r_{MU}})^{r_{P_\bullet}}$. Then it computes a message $M_{P_\bullet} \leftarrow \{ID_{P_\bullet} \| G_a \| g^{r_{P_\bullet}} \| alias \| G_h | g^{r_{MU}}\}$ and its signature $\sigma_{P_\bullet} \leftarrow \mathsf{Sig}(gpk_a, gsk_{P_\bullet}, M_{P_\bullet})$. Finally, the PoA replies CONNECTION REPLY $\leftarrow \{M_{P_\bullet}, \sigma_{P_\bullet}\}$ to the MU.

Step S3. When the MU receives the reply, it checks whether gpk_a is in its *keychain*. If so, the MU updates the time stamp on gpk_a to the current time. Otherwise, the MU reaches its CA (CA_h) to ask for gpk_a by sending a KEY QUERY $\leftarrow \{G_h, G_a, r_k\}$ to the PoA where r_k is a randomly selected nonce. The PoA finds out that the first group ID in the query (G_h) is not the group it belongs, so it relays the KEY QUERY to the CA of that group (CA_h).

Step S4. CA_h generates $M_k \leftarrow \{G_h \| G_a \| gpk_a \| r_k\}$, its signature $\sigma_k \leftarrow \mathsf{Sig}(gpk_h, gsk_{CA_h}, M_k)$, and replies a KEY ANSWER $\leftarrow \{M_k, \sigma_k\}$ to the PoA. The PoA relays it to the MU.

Step S5. The MU verifies the KEY ANSWER by checking if both $\mathsf{Vrf}(gpk_h, NULL, \sigma_k, M_k) = 1$ and $\mathsf{Trc}(gpk_h, stk_{CA_h}, \sigma_k, M_k) = 1$. If both are valid and nonce r_k is correct, the MU retrieves gpk_a from the KEY ANSWER. Otherwise, connection fails. If the *keychain* of the MU is full, it discards the key with the oldest time stamp. The MU adds gpk_a in its *keychain* and sets the time stamp to the current time.

Then, the MU verifies the signature in CONNECTION REPLY by checking if $\mathsf{Vrf}(gpk_a, NULL, \sigma_{P_\bullet}, M_{P_\bullet}) = 1$. If the signature is invalid or the nonce $g^{r_{MU}}$ is incorrect, the MU ignores the reply and the connection fails. Otherwise, the MU computes the session key $K \leftarrow (g^{r_{P_\bullet}})^{r_{MU}}$, a message $M_{MU} \leftarrow \{G_h \| alias \| g^{r_{P_\bullet}} \| g^{r_{MU}}\}$, its signature $\sigma_{MU} \leftarrow \mathsf{Sig}(gpk_h, gsk_{MU}, M_{MU})$, and sends a CONNECTION CONFIRM $\leftarrow \{M_{MU}, \sigma_{MU}\}$ to the PoA.

After the PoA receives the confirmation it verifies the signature by checking if $\mathsf{Vrf}(gpk_h, RL_h, \sigma_{MU}, M_{MU}) = 1$. If the signature is invalid or the nonce $g^{r_{P_\bullet}}$ is incorrect, the connection fails. Otherwise, the connection is established and both sides possess identical session keys.

C. Intra- and Inter-domain Handover Authentication Procedures

When an MU moves from an old serving PoA (P_b) to a new PoA (P_a), the MU executes the DAA intra- or inter-domain Handover Authentication procedure. As Fig. 3 shows, the details are

Steps H1 and H2. These steps are same as Steps S1 and S2.

Step H3. When the MU receives the reply, it checks if the new PoA is in the same group as the old one ($G_a \overset{?}{=} G_b$.) If so, the MU recognizes that this is an intra-domain handover authentication.

Step H4a. This step is similar to Step S5 except the key verification part is not executed in this step.

Note that in Step H3, if the new PoA is in the different group from the old one, the MU recognizes that this is an inter-domain handover authentication and following steps are executed.

Step H4b. The MU checks whether gpk_a is in its *keychain*. If so, the MU updates the time stamp on gpk_a to the current

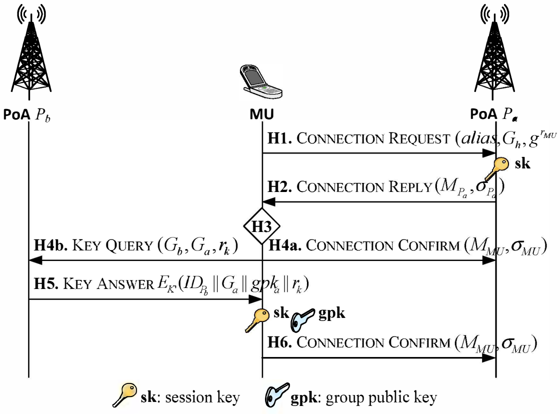
PoA P_b
MU
PoA P_a
H1. CONNECTION REQUEST $(alias, G_h, g^{r_{MU}})$
sk
H2. CONNECTION REPLY (M_{P_a}, σ_{P_a})
H3
H4b. KEY QUERY (G_b, G_a, r_k)
H4a. CONNECTION CONFIRM (M_{MU}, σ_{MU})
H5. KEY ANSWER $E_{K'}(ID_{P_b} \| G_a \| gpk_a \| r_k)$
sk
gpk
H6. CONNECTION CONFIRM (M_{MU}, σ_{MU})
sk: session key
gpk: group public key

Fig. 3. Message fl w of the DAA protocol in intra- and inter-domain handover authentication

time. Otherwise, the MU reaches the old PoA (P_b) to ask for gpk_a by sending KEY QUERY $\leftarrow \{G_b, G_a, r_k\}$ where r_k is a randomly selected nonce.

Step H5. P_b receives the query and find out that the firs group ID in the query (G_b) is the group it belongs, so it sets KEY ANSWER $\leftarrow E_{K'}(ID_{P_b} \| G_a \| gpk_a \| r_k)$, where E is a pre-stipulated symmetric encryption function and K' is the session key used between the MU and P_b. P_b then sends the KEY ANSWER back to the MU.

Step H6. The MU decrypts the message and checks if r_k is correct. If so, the MU retrieves gpk_a from the KEY ANSWER. Otherwise, connection fails. If the *keychain* of MU is full, it discards the key with the oldest time stamp. The MU adds gpk_a in its *keychain* and sets the time stamp to the current time. The following operations are similar to that in Step S5 except the key verificatio part.

V. SECURITY AND PERFORMANCE ANALYSIS

A. Security Analysis

As a network access control protocol utilizing asymmetric cipher, we can analyze the security of our DAA protocol base on following three criteria:

Mutual Authentication and Session Key Sharing. The unforgeable signatures in the messages fl w between MU and PoA provide secure authentication mutually. The unforgeability of signatures can be measured by the traceability game designed by Boneh and Shacham in [14], which is proved to be secure under q-Strong Diffie Hellman Assumption.The nonce in every message further protects the scheme against replay attack. In addition, the revocation list and Trc() procedure imbedded in the signature verificatio guarantee the validity of subscribers so that the scheme can be free from false data attacks initiate by outsiders and revoked members. Ephemeral session keys are established in each connections between MU and PoA via Diffi-Hellma key exchange procedure. Therefore, man-in-the-middle attacks can be prevented and perfect forward secrecy is achieved.

TABLE I
COMPARISON OF SECURITY AMONG AUTHENTICATION PROTOCOLS

Security features	DAA	Certificate based	IBC-based	Token-based
Mutual authentication	Yes	Yes	Yes	Yes
Anonymity against eavesdrop	Yes	Yes	Yes	Yes
Anonymity against PoA	Yes	No	No	No
Unlinkability	Yes	No	No	No
Non-repudiation	Yes	Yes	Yes	No
Resist fake PoA attack	Yes	No	No	No
Support handover	Yes	No	Yes	Yes

Integrity of Public Key Distribution. Public key distributions occur in session authentications and inter-domain handover authentications. In session authentications, MU gains new public keys from its CA via a foreign PoA. The integrity of the key is guaranteed by the signature of CA attached to the message in every KEY ANSWER. The signature is proved to be unforgeable and the nonce set by MU provides protection against replay attack. MU can confir that the KEY ANSWER is issued by its CA rather then anyone else in its group by activating Trc() with the signer tracing key of CA, which was given to every member when it joined the group.

In inter-domain handover authentications, MU gains new public keys from the previous PoA it has built connection with. The key is encrypted by the session key that is exclusively known by MU itself and the trusted PoA. Thus the integrity of the key is guaranteed. Also, nonce is used to prevent replay attack in the KEY QUERY.

User Anonymity With Unlinkability. Due to the usage of $alias$ in the initial CONNECTION REQUEST, user anonymity is provided against both eavesdroppers and foreign servers. However, on purpose of accounting and law disputes, the user anonymity can be broken by the signer tracing key which is exclusively possessed by MU itself and its group founder (CA of the group). The anonymity cannot be broken without $gmsk$. This property can be measured by the selfless-ano ymity game designed in [14], which is proved to be secure under Linear Assumption [11]. In addition, unlinkability is also proved to be achieved in the game so that PoA cannot link the two different connections initiated by the same unknown MU to build any user profile

Table I compares our scheme with the major distributed authentication schemes proposed for wireless network.

B. Performance Analysis

We analyze the performance of this protocol with respect to the computation overhead of authentication and communication overhead caused by number of messages exchanged.

Computation Overhead. Due to the limited storage space and battery power, computation overhead on MU should be minimized. The main operations of DAA and other exiting protocols are summarized in Table II.

TABLE II
COMPARISON OF COMPUTATION OVERHEAD AMONG AUTHENTICATION PROTOCOLS FOR DIFFERENT SCENARIOS IN MU

Scenario	Session	Intra-handover	Inter-handover
DAA	3 × G.Sig + 3 × G.Ver + DH†	2 × G.Sig + DH	2 × G.Sig + DH†
Certificate-base	2 × Asym + A.Ver + Sym		
IBC-based	I.Sig + I.Ver + DH		
Token-based	2 × Asym	0	0

† Amortized result.

TABLE III
COMPARISON OF COMMUNICATION OVERHEAD AMONG AUTHENTICATION PROTOCOLS FOR DIFFERENT SCENARIOS

Scenario	Session	Intra-handover	Inter-handover
DAA	3†	3	3†
Certificate-base	≥ 3	3	≥ 3
IBC-based	≥ 8	≥ 6	≥ 6
Token-based	3	5	5

† Amortized result.

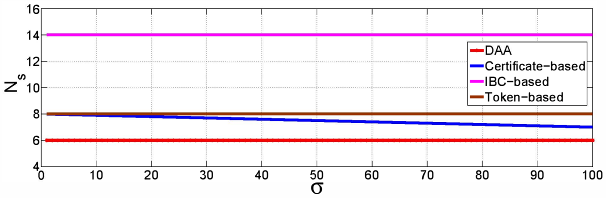

Fig. 4. Effects of ratio between number of inter-domain handovers over that of intra-domain handovers on communication overheads

In this table, X.Sig and X.Ver denote sign and verify processes of scheme X, where X can be G for group signature, A for traditional asymmetric cryptography, or I for IBC. Sym and Asym denote symmetric and asymmetric cryptography. DH denotes Diffie-Hellman key exchange. For our DAA protocol under session and inter-handover scenarios, the distribution of public key should cost an extra G.Ver and an extra Sym operation, respectively. With *keychain*, public key queries are unlikely to occur so the computation cost can be amortized down to the cost similar to intra-handover authentication.

Communication Overhead. Table III compares DAA with existing authentication protocols in terms of numbers of message flows. Although the distribution of group public keys costs extra messages (4 in session and 2 in inter-domain handover) in DAA, the *keychain* in our design amortizes the number of message flows per authentication under all three scenarios to approximately three flows. As shown in Fig. 4, DAA outperforms other distributed authentication protocols in communication loads in different ratios between number of inter-domain handovers over that of intra-domain handovers (denoted as ρ) on communication overheads (denoted N_s).

VI. CONCLUSION

Recently, the access authenticity and user privacy as mobile users (MUs) connect to heterogeneous radio access technologies receive attention. To resolve the problems of heavy signaling overhead and long signaling delay in centralized architecture, this paper proposed a novel distributed anonymous authentication (DAA) protocol utilizing group signature algorithms. By applying the MU and point of attachment (PoA) as group members, the adopted group signature algorithms provide identity verification directly without nodes sharing secrets in advance, which significantly reduces signaling overhead. Via group signature, the subjects of authentication are raised from nodes to groups not only to reinforce the protection against intruders, but also to provide user anonymity and unlinkability against foreign domains. Therefore, security and user privacy level are elevated to be suitable for nowadays heterogeneous networks. Performance analysis confirms advantages of DAA over existing solutions.

REFERENCES

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

[2] W. Liang and W. Wang, "A quantitative study of authentication and QoS in wireless IP networks," in *Proc. IEEE INFOCOM 2005*, vol. 2, Mar. 2005, pp. 1478–1489.

[3] G. Yang, D. S. Wong, and X. Deng, "Formal security definition and efficient construction for roaming with a privacy-preserving extension," in *J. Universal Comput. Sci.*, vol. 14, no. 3, 2008, pp. 441–462.

[4] K. Bayarou, M. Enzmann, E. Giessler, M. Haisch, B. Hunter, M. Ilyas, S. Rohr, and M. Schneider, "Towards certificate-based authentication for future mobile communications," *Wirel. Pers. Commun.*, vol. 29, no. 3-4, pp. 283–301, June 2004.

[5] M. Long, C. H. J. Wu, and J.D.Irwin, "Localized authentication for wireless LAN internetworking roaming," in *Proc. IEEE WCNC 2004*, vol. 1, Mar. 2004, pp. 264–267.

[6] J. M. Jeong, G. Y. Lee, and Y. Lee, "Mutual authentication protocols for the virtual home environment in 3G mobile network," in *Proc. IEEE Global Telecommunications Conference.*, vol. 2, Nov. 2002.

[7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Adv. Cryptology - Crypto, Springer LNCS*, vol. 196, Aug. 1984, pp. 47–53.

[8] D. Boneh and M. franklin, "Identity-based encryption from the weil pairing," in *Proc. Adv. Cryptology - Crypto, Springer LNCS*, vol. 2139, 2001, pp. 213–229.

[9] Y. Fu, J. He, R. Wang, and G. Li, "Mutual authentication in wireless mesh networks," in *Proc. IEEE ICC 2008*, May 2008, pp. 1690–1694.

[10] S. Machiraju, H. Chen, and J. Bolot, "Distributed authentication for low-cost wireless networks," in *Proc. ACM HotMobile 2008*, Feb. 2008, pp. 55–59.

[11] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Adv. Cryptology - CRYPTO 2004*, Oct. 2004, pp. 41–55.

[12] X. Lin, X. Sun, P. H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Trans.Vehic. Tech.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[13] D. Boneh and X. Boyen, "Short signatures without random oracles," in *EUROCRYPT 2004, Springer LNCS*, vol. 3027, 2004, pp. 56–73.

[14] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proc. ACM conference on Computer and Communications Security*, 2004, pp. 168–177.