Journal of Information Processing Vol. 18 63–76 (Mar. 2010)

Regular Paper

Anonymous IEEE802.1X Authentication System Using Group Signatures

Amang Sudarsono,[†1] Toru Nakanishi,[†1]
Yasuyuki Nogami[†1] and Nobuo Funabiki[†1]

Recently, ubiquitous Internet-access services have been provided by Internet service providers (ISPs) by deploying wireless local area networks (LANs) in public spaces including stations, hotels, and coffee shops. The IEEE802.1X protocol is usually used for user authentications to allow only authorized users to access services. Then, although user personal information of access locations, services, and operations can be easily collected by ISPs and thus, their strict management has been demanded, it becomes very difficult when multiple ISPs provide roaming services by their corporations. In this paper, we present an anonymous IEEE802.1X authentication system using a group signature scheme to allow user authentication without revealing their identities. Without user identities, ISPs cannot collect personal information. As an efficient revocable group signature scheme, we adopt the verifier-local revocation (VLR) type with some modifications for use of the fast pairing computation. We show the implementation of our proposal and evaluation results where the practicality of our system is confirmed for up to 1,000 revoked users.

1. Introduction

1.1 Backgrounds

Recently, mobile access to the Internet through wireless networks has become popular due to advancements of mobile communication devices such as mobile phones, personal digital assistants (PDAs), and laptop personal computers (PCs). To access the Internet, the user must make a contract to the ISP beforehand where the user access-code, such as the user ID and password, are given. By using this access-code, the user is authenticated as a privileged user of the ISP. Then, the ISP allows the user to connect to the wireless network services. Currently, the IEEE802.1X [24),26)–29)] is widely-used as the authentication protocol in wireless networks.

This protocol allows the ISP to collect the user information of the operations during connecting to networks by the access logs including the access locations, the access destinations, and the receiving services. Because this record is the user's sensitive private information, it is strongly required that the ISP manages it secretly. Unfortunately, there have lots of incidents that private information is leaked by insiders. This means that secure management of the private information is not easy.

As an ISP service for public wireless networks, the roaming is important. This means that a user of an ISP can use services by cooperative ISPs. In the roaming, the ISPs have to cooperatively manage the ID list of users and keep the access log secretly. As more ISP services take part in the roaming, the management becomes more complex so that the risk of the leakage of the private information becomes serious.

1.2 Our Contributions

In this paper, we present an anonymous IEEE802.1X authentication system for wireless networks using a group signature scheme. The group signature scheme [3)–10),12)–15)] is one of the anonymous authentication technology that allows each member of a group to sign messages on behalf of a group without revealing his own identity. In this scheme, a group manager (GM) has the authority to control the membership of members. In case of dispute, only a designated party can cancel the anonymity of a signature to trace and identify the signer. In our proposed system, the group may consist of users of an ISP service. The user sends the group signature to the authentication server of the ISP. Then, the user can prove that he is a valid user in the group without revealing the identity. The benefit is that the authentication server does not need the ID management and the secret keeping of the access log. On the other hand, since a strictly managed server can identify the user from the access log containing group signatures, the ISP can have responsibility of tracing a user who abuses the ISP services.

User revocations can often happen in the authentication system by a key loss, a stolen key, or voluntary leaving from services. Due to the anonymity, the revocation is actually not easy in the group signatures. Thus, the revocable group signature schemes have been proposed [5),7),9),10),12)–14)]. One type schemes [5),7),12),14)]

†1 Department of Communication Network Engineering, Okayama University

achieve the efficient performances, but require that signers fetch a revocation list with $O(R)$ size before signing, where R is the number of the revoked members. In the authentication system, a lots of user revocations happen, and thus R tends to be large. In addition, network connections are unstable. In the mobile communications, this type of revocation is not suitable for our authentication system. The other type of scheme is *Verifier-Local Revocation* (*VLR*)[9),10),13)]. Because the signers do not fetch the revocation list, it is suitable for our system in the mobile environments. Among of the VLR group signature schemes, the pairing-based scheme Ref. 11) achieves the shortest signatures with the strong anonymity. On the other hand, according to Refs. 22) and 23), there are two types of asymmetric pairing on bilinear groups $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$; asymmetric pairings for which an efficiently-computable homomorphism between $\mathcal{G}_1$ and $\mathcal{G}_2$ is known are called type-2 pairings and asymmetric pairings for which no efficiently-computable homomorphism is known between $\mathcal{G}_1$ and $\mathcal{G}_2$ are called type-3 pairings. Scheme Ref. 11) has to use type-2 pairings, due to the use of such a homomorphism. However, in case of using type-2 pairings, testing membership elements on $\mathcal{G}_2$ need heavy computations in the implementation of the state-of-the-art Barreto-Naehrig curves [16)], which is shown in Ref. 23) (Table 2), and thus the verification with the test also needs lots of costs. On the other hand, in case of using type-3 pairings, the anonymity proof cannot be proved due to the lack of the homomorphism. In this paper, we modify the previous scheme [10)] to have the provable anonymity without the homomorphism, and implemented the modified scheme using an efficient type-3 pairing [16),19)].

We implemented the authentication system by extending *EAP-TTLS* [29)] in the IEEE802.1X protocol such that it can use the group signature as the digital certificate of the client. EAP-TTLS is a protocol to authenticate the server by the digital certificate and the user by ID and password. Then, we evaluate authentication times to show the effectiveness of our system.

This paper is organized as follows: Section 2 describes the overview of the IEEE802.1X authentication protocol and the model of VLR group signature scheme as the preliminaries of our proposed system. Section 3 describes the modified VLR group signature scheme for anonymous IEEE802.1X authentication. Section 4 proposes an anonymous IEEE802.1X authentication system. Our implementation and the experimental results are discussed in Section 5 and Section 6, respectively. The conclusion of this paper and future works are discussed in Section 7.

2. Preliminaries

Our proposed anonymous IEEE802.1X authentication system is derived from the IEEE802.1X authentication protocol and the VLR group signature scheme. In this section, we describe the overview of the IEEE802.1X authentication protocol and the model of the VLR group signature scheme.

2.1 Overview of IEEE802.1X Authentication Protocol

The IEEE802.1X protocol provides port-based authentication, which involves communications between a supplicant, authenticator, and authentication server. The supplicant is often software on a client, such as laptop PC, the authenticator is a wired Ethernet switch or wireless AP, and the authentication server is generally a RADIUS. For communications between the mobile host and the AP, and between the authentication server and the AP, the *Extensible Authentication Protocol* (*EAP*) [26)] and the RADIUS protocol are used, respectively.

Based upon EAP, the IEEE802.1X protocol can use several authentication methods, such as *Message Digest 5* (*MD5*), *Transport Layer Security* (*TLS*), *Tunneled TLS* (*TTLS*), *Protected Extensible Authentication Protocol* (*PEAP*), and *Lightweight Extensible Authentication Protocol* (*LEAP*). Among these authentication methods, the following methods have mainly been used in real worlds. EAP-MD5 [26)] authenticates the user by his ID and password after hashing them by the MD5 hash function. EAP-TLS [27)] mutually authenticates both the user and the server by using digital certificates. EAP-TTLS and EAP-PEAP authenticate the server by the digital certificate and the user by ID and password. The EAP-TTLS extends EAP-TLS to exchange messages between the client and the server by using the secure tunnel established by the TLS [28)] protocol. Among these methods, we adopt the EAP-TTLS as the basis of our authentication protocol, since it offers the server authentication and allows our anonymous authentication to be easily integrated. The protocol flow of the EAP-TTLS is shown in **Fig. 1**.

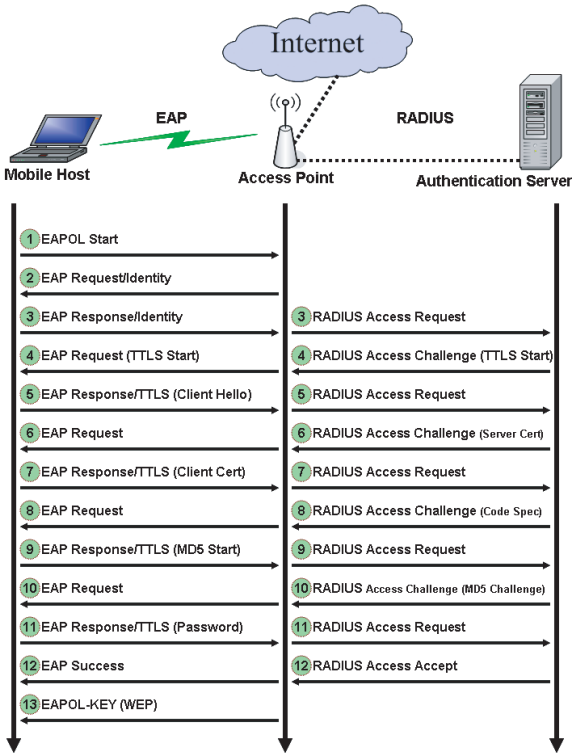

Fig. 1 Protocol flow of EAP-TTLS.

2.2 Model of VLR Group Signature Scheme

The VLR group signature scheme consists of the following algorithms. In the VLR schemes, the time is divided into intervals [10].

Setup: It is a probabilistic algorithm on inputs n, which is the number of members, and T, which is the number of time intervals. It outputs a group public key gpk, a GM's secret key $gmsk$, and a tracing secret key tsk. The tracing key tsk is used to trace the actual signer from the group signature.

Join: It is an interactive protocol between a probabilistic algorithm **Join-U** for the i-th user and a probabilistic algorithm **Join-GM** for GM. **Join-U** on input gpk, outputs $msk[i]$ that is the i-th user's secret key. **Join-GM**, on inputs gpk, $gmsk$ and a group member list GL, renews GL.

Revoke: It is a probabilistic algorithm on inputs gpk, a time interval j, a set of revoked users RU and the member list GL. The output is a revocation list $RL[j]$ that consists of revocation tokens for the revoked users at j.

Sign: It is a probabilistic algorithm which takes as inputs gpk, the current time interval j, a secret key $msk[i]$, and a message $M \in \{0,1\}^*$, and the output is a group signature σ.

Verify: It is a deterministic algorithm for verification on inputs gpk, the revocation list $RL[j]$ at the time interval j, a signature σ, and the message M. Then, it outputs either "valid" or "invalid". The validity means that σ is a correct signature on M at interval j w.r.t gpk, and that the signer is not revoked at the time interval j.

Open: It is a deterministic algorithm performed by a party (Opener) to trace the actual signer on inputs gpk, the signature σ of the traced user, the tracing secret key tsk, and a list of group members GL. Then, it outputs the user identifier i.

A secure VLR group signature scheme must satisfy the following properties.

1. **Unforgeability:** no one except group members is able to generate a valid signature.
2. **Anonymity:** given a signature, no one except the signer and the opener is able to identify the signer.
3. **Unlinkability:** no one except the signer and the opener can determine whether two different signatures were generated by the same signer or not.
4. **Traceability:** in case of dispute, the opener is able to successfully identify the actual signer.

These are informal definitions. The formal definitions are described in Ref. 10).

3. Modified VLR Group Signature Scheme for Anonymous Authentication

In this section, we present the modified VLR group signature scheme for our authentication system to prove the anonymity without the homomorphism.

3.1 Bilinear Maps

Our group signature scheme utilizes the following bilinear groups:

1. $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_T$ are multiplicative cyclic groups of prime order p,
2. g_1 and g_2 are randomly chosen generators of $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively.
3. e is an efficiently computable bilinear map: $\mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$, i.e., **(a)** for all $u, u' \in \mathcal{G}_1$ and $v, v' \in \mathcal{G}_2$, $e(uu', v) = e(u, v)e(u', v)$ and $e(u, vv') = e(u, v)e(u, v')$, and **(b)** $e(g_1, g_2) \neq 1$.

The bilinear map can be efficiently implemented with the pairings. There are two types of bilinear pairings, symmetric ($\mathcal{G}_1 = \mathcal{G}_2$) and asymmetric ($\mathcal{G}_1 \neq \mathcal{G}_2$). The symmetric pairings also can be called as type-1 pairings [22),23)]. As commented in Ref. 23), at the 128-bit security level, the asymmetric type is faster than the symmetric type. Thus, we concentrate on the asymmetric type. There are two types of asymmetric pairing on bilinear groups $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$; asymmetric pairings for which an efficiently-computable homomorphism between $\mathcal{G}_1$ and $\mathcal{G}_2$ is known are called as type-2 pairings and asymmetric pairings for which no efficiently-computable homomorphism is known between $\mathcal{G}_1$ and $\mathcal{G}_2$ are called type-3 pairings [22),23)].

3.2 Assumptions

As well as the original scheme [10)], the unforgeability and traceability requirements of our implemented scheme are based on the q-SDH assumption [12)]. Since this paper does not address the requirements, the definition of this assumption is omitted. On the other hand, we modify the DLIN assumption [10),12)] to the modified DLIN assumption, as follows.

Definition 1 (Decision Linear (DLIN) assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$|\Pr[\mathcal{A}(u, v, w, u^a, v^b, w^{a+b}) = 0] - \Pr[\mathcal{A}(u, v, w, u^a, v^b, w^c) = 0]| \quad (1)$$

is negligible, where $u, v, w \in_R \mathcal{G}_2$ and $a, b, c \in_R \mathbb{Z}_p^*$.

Definition 2 (Modified DLIN assumption) For all PPT algorithm $\mathcal{A}$, the probability

$$|\Pr[\mathcal{A}(u, v, w, \tilde{u}, \tilde{v}, \tilde{w}, u^a, v^b, w^{a+b}) = 0] - \Pr[\mathcal{A}(u, v, w, \tilde{u}, \tilde{v}, \tilde{w}, u^a, v^b, w^c) = 0]| \quad (2)$$

is negligible, where $u \in_R \mathcal{G}_1$, $\tilde{u} \in_R \mathcal{G}_2$, $a, b, c, \rho_1, \rho_2 \in_R \mathbb{Z}_p^*$, and $v = u^{\rho_1}, w = u^{\rho_2}, \tilde{v} = \tilde{u}^{\rho_1}, \tilde{w} = \tilde{u}^{\rho_2}$.

In Ref. 12), the hardness of the DLIN assumption in the generic group model is proved in the setting with the isomorphism between $\mathcal{G}_1, \mathcal{G}_2$. The additional values $\tilde{u}, \tilde{v}, \tilde{w}$ in the modified DLIN assumption can be generated from u, v, w using this isomorphism. Thus, the proof in Ref. 12) can be applied to the modified DLIN assumption, namely we can prove the hardness of the modified DLIN assumption in the generic group model.

To adopt the tracing mechanism of Ref. 15), we also need the DDH assumption.

3.3 Modified VLR Group Signature Scheme

3.3.1 Construction Idea

The underlying scheme is the VLR scheme of Ref. 11) with the shortest signatures among the VLR types. The scheme must use a type-2 pairing because of employing a homomorphism between $\mathcal{G}_1$ and $\mathcal{G}_2$ to prove the anonymity. However, in case of using type-2 pairings, testing membership elements on $\mathcal{G}_2$ need heavy computations in the implementation of the state-of-the-art Barreto-Naehrig curves [16)], which is shown in Ref. 23) (Table 2). On the other hand, in case of using type-3 pairings, the anonymity proof cannot be proved due to the lack of the homomorphism. Therefore, we will modify the scheme without the homomorphism dependencies to adopt a type-3 pairing with very efficient pairing implementation.

The revocation mechanism in the underlying scheme is as follows. Let ψ be the homomorphism from $\mathcal{G}_2$ to $\mathcal{G}_1$. At each interval j, GM publishes the revocation token $B_{ij} = \psi(h_j)^{y_i}$, where $h_j \in \mathcal{G}_2$ is a public value corresponding to the interval j, and y_i is a secret value of the revoked member i. On the other hand, the signature of the member i includes a revocation tag $T_2 = \psi(f)^{\beta+y_i}$, $T_3 = \psi(h_j)^{\beta}$, where $f \in \mathcal{G}_2$ is a hashed random value, and β is a random secret. Then, any verifier can check $e(T_2, h_j) \stackrel{?}{=} e(B_{ij}T_3, f)$, which holds if and only if the signer is the same as the target member of the revocation token. In the anonymity proof, the inputs of $\mathcal{G}_2$ in the DLIN assumption are transformed via ψ to the corresponding $\mathcal{G}_1$ elements and the signature and the revocation tokens are simulated. However, without ψ, the anonymity proof (and the scheme) is not valid.

In this paper, we modify the scheme and the anonymity proof without ψ. In the modified scheme, we utilize public values $(\hat{h_j}, h_j) = (g_1^{r_j}, g_2^{r_j}) \in (\mathcal{G}_1, \mathcal{G}_2)$ for

$r_j \in_R \mathbb{Z}_p^*$, instead of $(\psi(h_j), h_j)$. Also, the signer utilizes $(\hat{f}, f) = (g_1^r, g_2^r) \in (\mathcal{G}_1, \mathcal{G}_2)$ for $r \in_R \mathbb{Z}_p^*$ instead of $(\psi(f), f)$. The computations of T_2 and T_3 and the check equation are the same.

In the following construction, we utilize the version with the exculpability requirement (protection against GM's forging), which is described in the paper version [11] of Ref. 10). In addition, we add the open algorithm to identify the actual signers from the group signatures, since the original scheme omits the algorithm. The tracing mechanism is the same as Ref. 15), where we need another group $\mathcal{G}$ with the same prime order p as the bilinear groups such that the DDH assumption on $\mathcal{G}$ holds. We can generate a certain *non pairing-friendly* curve of the prime order p by complex multiplication method [1].

3.3.2 *SPK*s

As well as Ref. 10), we adopt signatures converted by Fiat-Shamir heuristic from zero-knowledge proofs of knowledge (PK). We call the signatures SPKs. The SPKs we adopt are the generalization of the Schnorr signature. We introduce the following notation, $SPK\{(x_1, \ldots, x_t) : R(x_1, \ldots, x_t)\}(M)$, which means that a signature of message M by a signer who knows secret values $x_1, \ldots, x_t$ satisfying a relation $R(x_1, \ldots, x_t)$. This paper utilizes an SPK proving the knowledge of a representation of $C \in \mathcal{G}_1$ to the bases $g_1, g_2, \ldots, g_t \in \mathcal{G}_1$ on message M, which is denoted as $SPK\{(x_1, \ldots, x_t) : C = g_1^{x_1} \cdots g_t^{x_t}\}(M)$. This can be also constructed on groups $\mathcal{G}_2$, $\mathcal{G}_T$, and $\mathcal{G}$. The SPK can be extended to proving multiple representations with equal parts.

3.3.3 Algorithms

The details of algorithms are described as follows.

Setup: Setup algorithm is given n, the number of members, and T, the number of time intervals and GM performs the following steps:

1. Select generators $g_1 \in \mathcal{G}_1$ and $g_2 \in \mathcal{G}_2$. Additionally, select $\hat{g}_1, \tilde{g}_1 \in_R \mathcal{G}_1$.
2. For all $j \in [1, T]$, select $r_j \in_R \mathbb{Z}_p^*$. Then compute $\hat{h}_j = g_1^{r_j}$ and $h_j = g_2^{r_j}$.
3. Select $\gamma \in_R \mathbb{Z}_p^*$, and compute $Y = g_2^{\gamma}$.
4. Select a generator $g \in \mathcal{G}$ and random numbers $s, t \in_R \mathbb{Z}_p^*$. Compute $S = g^s$ and $T = g^t$.
5. Output the group public key $gpk = (g, g_1, g_2, \hat{g}_1, \tilde{g}_1, Y, \hat{h}_j, h_j, S, T, p, \mathcal{G}, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, e)$, GM's secret key $gmsk = (\gamma)$, and the tracing secret key $tsk = (s, t)$.

Join: The i-th user (Member i) joins the group as follows.

1. Member i selects $x_i, z_i' \in_R \mathbb{Z}_p^*$, and computes $H_i = \hat{g}_1^{x_i} \tilde{g}_1^{z_i'}$ and $Q_i = g^{x_i}$.
2. Member i sends GM H_i, Q_i and proves $H_i = \hat{g}_1^{x_i} \tilde{g}_1^{z_i'}$ and $Q_i = g^{x_i}$ by an SPK.
3. GM chooses y_i, $z_i'' \in_R \mathbb{Z}_p^*$, and computes $A_i = (g_1 H_i \tilde{g}_1^{z_i''})^{1/(\gamma + y_i)}$. GM sends A_i, y_i, z_i'' to member i. GM adds (i, y_i, Q_i) to GL.
4. Member i computes $z_i = z_i' + z_i''$, and outputs $msk[i] = (A_i, x_i, y_i, z_i)$.

Revoke: The input of this algorithm are gpk, the revocation token at interval j of revoked users RU which includes $y_1, \ldots, y_R$ elements, where $R = |R|$. The algorithm is as follows.

1. For each revoked members i' in RU, compute $B_{i'j} = \hat{h}_j^{y_{i'}}$, by fetching $y_{i'}$ from GL.
2. Output a revocation list $RL[j]$ that consists of all $B_{i'j}$.

Sign: The inputs of this signing algorithm are group public key gpk, the signer's secret key $msk[i] = (A_i, x_i, y_i, z_i)$, the current time interval j, and a signed message $M \in \{0,1\}^*$. We assume that M includes the time interval j in order to bind the signature to the interval. The algorithm is as follows:

1. Select $r \in_R \mathbb{Z}_p^*$. Compute $\hat{f} = g_1^r$, $f = g_2^r$.
2. Select $\alpha, \beta \in_R \mathbb{Z}_p^*$ and set $\zeta = z_i - \alpha y_i$. Compute $T_1 = A_i \tilde{g}_1^{\alpha}$, $T_2 = \hat{f}^{\beta + y_i}$, and $T_3 = \hat{h}_j^{\beta}$.
3. Select $u \in_R \mathbb{Z}_p^*$. Compute $U = g^{x_i + u}$, $V = S^u$, and $W = T^u$.
4. The SPK X is computed as follows.

$$
\begin{aligned}
X = SPK\{&(x_i, y_i, \alpha, \beta, \zeta, r, u) : \\
&e(g_1, g_2)/e(T_1, Y) = e(T_1, g_2)^{y_i} e(\hat{g}_1, g_2)^{x_i} e(\tilde{g}_1, g_2)^{\zeta} e(\tilde{g}_1, Y)^{-\alpha} && (3)\\
&\wedge T_2 = \hat{f}^{\beta + y_i} \wedge T_3 = \hat{h}_j^{\beta} \wedge \hat{f} = g_1^r \wedge f = g_2^r && (4)\\
&\wedge U = g^{x_i + u} \wedge V = S^u \wedge W = T^u\}(M). && (5)
\end{aligned}
$$

This SPK is the same as the original [10),11)], except the correctness of $\hat{f}, f$, and U, V, W. Note that in the Refs. 10) and 11), f value is derived from $f = H_0(gpk, M, r)$, where H_0 is a hash function with respective range $\mathcal{G}_2$ and

r is a random nonce which are treated as random oracles. The values U, V, W are a ciphertext for tracing, which is the same as Ref. 15). Concretely, the computation of the $SPK\ X$ is as follows.

a. Pick blinding factors r_{x_i}, r_{y_i}, r_α, r_β, r_ζ, r_r, $r_u \in_R \mathbb{Z}_p^*$.

b. Compute

$$R_1 = e(T_1, g_2)^{r_{y_i}} e(\hat{g}_1, g_2)^{r_{x_i}} e(\tilde{g}_1, g_2)^{r_\zeta} e(\tilde{g}_1, Y)^{-r_\alpha}, \quad (6)$$
$$R_2 = \hat{f}^{r_\beta + r_{y_i}}, R_3 = \hat{h}_j^{r_\beta}, \quad (7)$$
$$R_4 = g_1^{r_r}, R_5 = g_2^{r_r}, \quad (8)$$
$$R_6 = g^{r_{x_i} + r_u}, R_7 = S^{r_u}, R_8 = T^{r_u}. \quad (9)$$

c. Compute a challenge $c \in \mathbb{Z}_p^*$ as $c = \mathcal{H}(gpk, j, M, T_1, T_2, T_3, \hat{f}, f, U, V, W, R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8)$.

d. Compute responses $s_{x_i} = r_{x_i} + cx_i$, $s_{y_i} = r_{y_i} + cy_i$, $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, $s_\zeta = r_\zeta + c\zeta$, $s_r = r_r + cr$, and $s_u = r_u + cu$.

5. Output the group signature $\sigma = (T_1, T_2, T_3, \hat{f}, f, U, V, W, c, s_{x_i}, s_{y_i}, s_\alpha, s_\beta, s_\zeta, s_r, s_u)$.

Verify: Inputs of this algorithm are gpk, the current time interval j, the revocation list $RL[j]$ for all revoked members i' at the interval j, a target signature σ and the message $M \in \{0,1\}^*$. The signature σ is verified as follows.

1. **Signature check**: Check that σ is valid, by checking the $SPK\ X$, as follows.

a. Rederive $\tilde{R}_1$, $\tilde{R}_2$, $\tilde{R}_3$, $\tilde{R}_4$, $\tilde{R}_5$, $\tilde{R}_6$, $\tilde{R}_7$, and $\tilde{R}_8$ as

$$\tilde{R}_1 = e(T_1, g_2)^{s_{y_i}} e(\hat{g}_1, g_2)^{s_{x_i}} e(\tilde{g}_1, g_2)^{s_\zeta} e(\tilde{g}_1, Y)^{-s_\alpha} \cdot (e(g_1, g_2)/e(T_1, Y))^{-c}, \quad (10)$$
$$\tilde{R}_2 = \hat{f}^{s_\beta + s_{y_i}} / T_2^c, \tilde{R}_3 = \hat{h}_j^{s_\beta} / T_3^c, \quad (11)$$
$$\tilde{R}_4 = g_1^{s_r} / \hat{f}^c, \tilde{R}_5 = g_2^{s_r} / f^c, \quad (12)$$
$$\tilde{R}_6 = g^{s_{x_i} + s_u} / U^c, \tilde{R}_7 = S^{s_u} / V^c, \tilde{R}_8 = T^{s_u} / W^c. \quad (13)$$

b. Rederive the challenge as $c' = \mathcal{H}(gpk, j, M, T_1, T_2, T_3, \hat{f}, f, U, V, W, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4, \tilde{R}_5, \tilde{R}_6, \tilde{R}_7, \tilde{R}_8) \in \mathbb{Z}_p^*$. Check that $c \stackrel{?}{=} c'$, which means that the signature is "valid" or "invalid".

2. **Revocation check**: Check that the signer is not revoked at the interval j, by checking:

$$e(T_2, h_j) \stackrel{?}{=} e(B_{i'j} T_3, f) \quad (14)$$

for all $B_{i'j} = \hat{h}_j^{y_{i'}} \in RL[j]$. The output is "valid" or "invalid".

Open: Inputs of this algorithm are gpk, the traced user's signature σ on message M, a tracing key tsk. The opener traces and identifies the signer as follows.

1. Verify the traced signature by using **Verify** algorithm.
2. If the signature is valid, compute $Q_i = U/V^{-s}$, using the tracing key s.
3. Output i.

The formal definition and proof of the anonymity are in Appendix.

3.4 Efficiency Consideration

To confirm the better efficiency of the proposed scheme using the type-3 pairings than the previous scheme using the type-2 pairings, we compare the efficiency of **Sign** and **Verify** algorithms, excluding **Join**-related and **Open**-related parts and the revocation check (these costs are the same between the previous and proposed schemes). The comparison considers the use of pre-computations of Ref. 11) for both schemes. Note that the previous scheme cannot employ the type-2 pairings without any modification. This is because the scheme uses hashing to $\mathcal{G}_2$, for which no efficient method is known in case of the type-2 pairings [22),23)]. Therefore, the element f in the previous scheme is obtained by computing $f = g_2^r$ instead of securely hashing to $\mathcal{G}_2$, although there could be better modification. Moreover, in the verification, testing membership on $\mathcal{G}_2$ for f is required. In this comparison, the previous scheme includes these modifications.

Table 1 shows the comparison of the computation costs.

As the overhead, the proposed scheme needs slightly more exponentiation on

Table 1 Comparisons of computation costs of **Sign** and **Verify**. {$E(\mathcal{G})$: exponentiation in $\mathcal{G} \in (\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$, P: pairing, $T(\mathcal{G}_2)$: testing membership on $\mathcal{G}_2$.}

Scheme	Computation costs of **Sign** Computation costs of **Verify**
10), 11)	$5E(\mathcal{G}_1) + E(\mathcal{G}_2) + 3E(\mathcal{G}_T)$ $4E(\mathcal{G}_1) + 2E(\mathcal{G}_2) + 2E(\mathcal{G}_T) + P + T(\mathcal{G}_2)$
Proposed Scheme	$7E(\mathcal{G}_1) + 2E(\mathcal{G}_2) + 3E(\mathcal{G}_T)$ $6E(\mathcal{G}_1) + 4E(\mathcal{G}_2) + 2E(\mathcal{G}_T) + P + T(\mathcal{G}_2)$

$\mathcal{G}_1$ and $\mathcal{G}_2$, although it has the same computational costs on $\mathcal{G}_T$ and pairings. Here, we concentrate on the $\mathcal{G}_1, \mathcal{G}_2$-related computation costs. Using Table 2 of Ref. 23), we can estimate the costs on the basis of the differences of operation costs in the type-2 and type-3 pairings. Note that, among the operations, testing membership in $\mathcal{G}_2$ needs much more costs in case of the type-2 pairings. Then, **Sign** and **Verify** of the previous scheme are 12,250 m and 39,077 m, respectively, where m means the cost of multiplication. While those of the proposed scheme are 16,835 m and 24,458 m, respectively. This result means that **Sign** of the proposed scheme is less efficient, although **Verify** is much more efficient. For **Sign**, all the heavy computations can be pre-computed before the message is decided. By taking account of the pre-computations, we conclude that the proposed scheme is more efficient.

3.5 Utilized Pairing Library

This implementation utilizes the pairing library [17)–21)] based on the GMP library [30)], and the group order is 254 bits and the embedding degree is 12 (Barreto-Naehrig curve [16)]). This pairing library gives the fast pairing called "*Cross-twisted* χ-based Ate (Xt-Xate) pairing" with *subfield-twisted* curve [18),19)]. The number of iterations of Miller's algorithm for the Xt-Xate pairing is about one-quarter of the plain Tate pairing. In addition, using efficiently-computable endomorphisms and isomorphisms, elliptic curve operations are accelerated [20),21)]. Thus, based on good properties of Barreto-Naehrig curve, this library exhaustively and totally accelerates not only pairings but also the other elliptic curve operations together with Gauss Period Normal Bases (GNB).

4. Proposed Anonymous IEEE802.1X Authentication System

In the proposed anonymous IEEE802.1X authentication system, there are 4 main entities: APs and authentication servers that are managed by an ISP, a registration server that are managed by another ISP, and mobile hosts as the users. **Figure 2** and **Fig. 3** illustrate the protocols of the proposed system and its behavior in the anonymous authentication system. The authentication server is the RADIUS server to authenticate the accessing mobile host. The registration server manages the group of users, allowing a new user to join the group and revoking the membership of users. In addition, the registration server has the authority to trace and identify an anonymous misbehaving user.

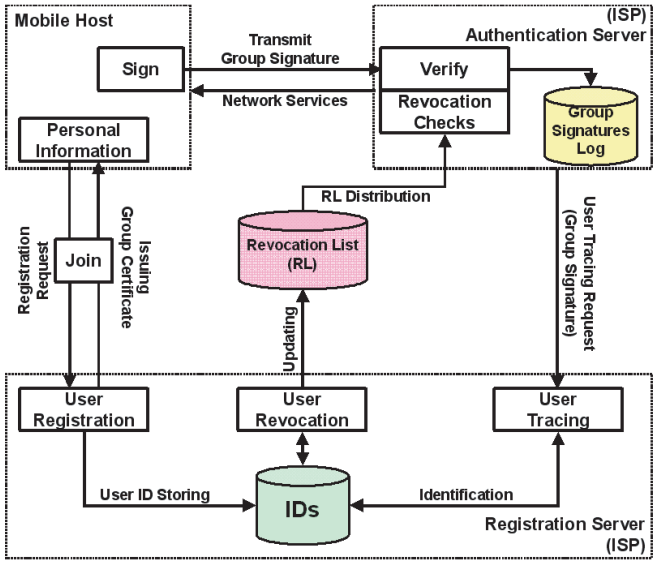

Fig. 2 Protocols of proposed system.

The system model is divided into five protocols as follows.

4.1 Setup

In advance, the registration server runs the setup algorithm of the VLR group signature scheme, and obtains keys. The group public key is distributed to the authentication servers.

4.2 Registration

This protocol is used to register the new user with the system. It consists of the following steps:

1. Registration of user personal information:
   The user is required to register his personal information with the registration server in order to use the ISP services.
2. Join protocol of group signature scheme:
   After obtaining the ID and the group public key of the ISP, the user executes the join protocol of the VLR group signature scheme, where the registration

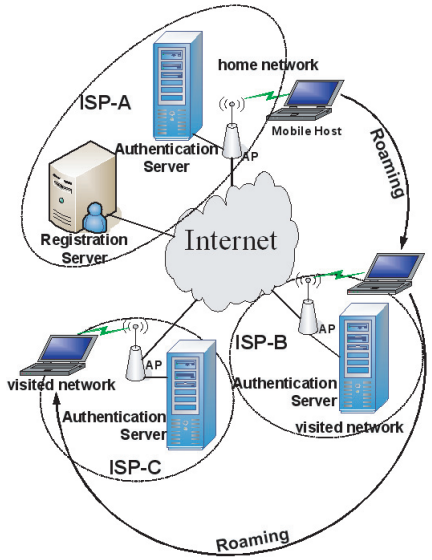

Fig. 3 Proposed anonymous authentication system.

server behaves as GM.

4.3 Authentication

This protocol is used to authenticate the user who wants to use network services. This protocol is divided into the following steps:

1. Generation and transmission of a group signature:
   The mobile host generates the group signature and sends it to the authentication server via the associated AP. Furthermore, the mobile host needs to send the ID of the ISP that the user registered to.
2. Verification of group signature:
   In the authentication server, the group signature from the mobile host is verified using the group public key corresponding to the ISP specified by the user. If the verification is correctly done, the connection to network is permitted.

4.4 Tracing

When a user misuses the anonymity during the Internet connections, the authentication server sends the group signature of this user to the registration server of the ISP. Then, the registration server can identify this user using the open algorithm.

4.5 User Revocation

This protocol is used when the registration server wants to revoke the user. The steps of this protocol are described as follows:

1. Revocation list update:
   The registration server updates the revocation list to reflect the revocation of the user.
2. Revocation list transmission:
   The latest revocation list is transmitted to the entire authentication servers.

4.6 Protocol Behaviors in Our Anonymous Authentication System

Here we discuss the behaviors of our protocols in the proposed anonymous IEEE802.1X authentication system.

In the authentication, the mobile host sends his group signature together with the ID of the registered ISP. The sent data includes no private information on the user, due to the anonymity and unlinkability of the group signature. Thus, dishonest parties, including insiders who can access to the authentication servers, cannot obtain any private information of users.

On the other hand, since only the registration server has the authority to identify the user from the signature in an access log, the ISP can have the responsibility to trace abusing users.

In addition, we have a user revocation mechanism to address leaving of services or the key loss. The mobile host does not need the revocation list, and the computation and communication costs are low. The evaluation based on the implementation is shown in Section 6.

Figure 3 shows a scenario of roaming ISP services for public wireless networks. A mobile host, which registers with the registration server of the ISP-A, can utilize not only APs of the ISP-A, but also APs of the cooperative ISP-B and ISP-C. Note that the ISP-B and ISP-C need not to manage the IDs of the users in the ISP-A. The authentication servers have only to verify the group signature sent from the mobile host.

4.7 Related Issues

Outside the authentication protocol, there are some issues related the anonymity. One is from the MAC address of users' wireless interfaces. The ISP can collect the MAC addresses of connected users. The MAC address is a unique hardware address of the network interface card and thus the ISP can link connections by the same user. In this system, we can employee a MAC changer tool to change the actual MAC address into a random MAC address and easily incorporate it to our supplicant software. Hence, every time the user connect to the network, he always uses a faked MAC address assigned by his supplicant software.

The other issue is from the radio signature in the physical layer. The issue and the countermeasure are discussed in Ref. 2), for example.

5. Implementation of Anonymous IEEE802.1X Authentication System

To show the effectiveness of our system, we implemented the main protocols: authentication protocol and the user revocation protocol.

5.1 Implementation of Authentication Protocol

To implement our authentication protocol, we utilize *WPA_Supplicant* and *FreeRADIUS*. The WPA_Supplicant is the open-source software implementation of the IEEE802.11i Supplicant for Linux, WINDOWS, etc. The FreeRADIUS is the open-source software implementation of the RADIUS server.

In our implementation, we adopt EAP-TTLS rather than EAP-TLS. In EAP-TTLS, any client authentication can be used in the secure tunnel with the server. On the other hand, EAP-TLS needs the complex certificate-based client authentication. Note that our current anonymous authentication protocol is different from the usual certificate-based authentication, since the certificate part is not needed. This is because we assume that the group public key is authenticated in advance. Thus, in EAP-TTLS, it is easier to integrate our authentication protocol. Furthermore, a better revocation method in the group signatures, where the revocation-related data has to be sent to the client from the server, may be invented. In that case, the EAP-TTLS can easily replace the authentication with the new method rather than EAP-TLS. These are the reason why we adopt EAP-TTLS.

We designed the new protocol, named EAP-TTLS/GS, by incorporating our VLR group signature scheme. The modified points from EAP-TTLS are STEP10, STEP11, and STEP12 of protocol flow of EAP-TTLS as shown in Fig. 1, where the other steps are the same as the original EAP-TTLS.

STEP10: EAP Request GS-Challenge/RADIUS Access-Challenge
A 128-bit random number is transmitted from the authentication server to the mobile host.

STEP11: EAP Response GS-Response/RADIUS Access-Request
The mobile host generates a group signature, where the received random number is used as the message to be signed. Then, the group signature is sent to the authentication server.

STEP12: EAP Success/RADIUS Access-Accept
Based on the random number and the latest revocation list from the registration server, the authentication server verifies the group signature. When the verification succeeds, the authentication server transmits the *RADIUS Access-Accept* packet to open the connection to the AP. Otherwise, the authentication server transmits the *RADIUS Access-Reject* packet to disable the connection.

5.2 Implementation of User Revocation

In the user revocation protocol, the registration server distributes the latest revocation list to the authentication server. For this purpose, we utilize *UNISON*[31] as a high speed file-synchronization tool. It allows two replicas of a collection of files and directories to be stored on different hosts over the network securely, since this tool can work over the encrypted SSH connection. By using this tool, we synchronize the file for the revocation list between the registration and the authentication servers. Thus, whenever the registration server updates the revocation list, it is automatically updated in the RADIUS servers. Whenever the revocation list is updated, the RADIUS servers always detect and capture it automatically.

6. Experimental Results

In this section, we present the experimental results to show the efficiency of

Table 2 Specifications of mobile host and authentication server in experiments.

Spec. of	Mobile host	Authentication server
Software	WPA_Supplicant-0.6.0 Openssl-0.9.8c gmp-4.2.1	FreeRADIUS-server-2.0.0 Openssl-0.9.8g gmp-4.2.2
O/S	Debian Linux kernel-2.6.18-6-686	Gentoo Linux kernel-2.6.24-r3
CPU	Intel Pentium(R)M 600 MHz	Intel Core(TM)2 Duo 2.66 GHz
RAM	256 MB	2 GB
NIC	Intel(R) PRO/Wireless 2200BG Network Connection	Realtek RTL8111/8168B PCI Express Gigabit Ethernet

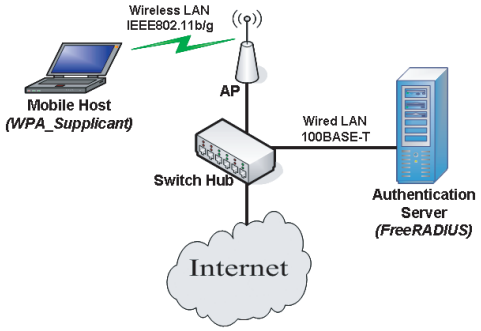

Fig. 4 Experimental environment.

the authentication protocol, which has a great impact on the practicality of our system. Additionally, we show the processing time of **open** algorithm.

6.1 Experimental Environment

We measured the performance in a laptop PC for the mobile host and a desktop PC for the authentication server. The specifications of these PCs are shown in **Table 2**. We used the mobile host with the low-speed performance to show the effectiveness in mobile environments. In fact, although the original CPU frequency of the mobile host is 1.1 GHz, we configured it to 600 MHz for this purpose. As the access point, we used Buffalo AirStation WAPM-HP-AM54G54 Access Point. **Figure 4** shows the experimental environment of our system. In our experiments, we vary the number of revoked users (R) from 40 to 1,000. This is because the authentication time depends on R, not on n that is the total

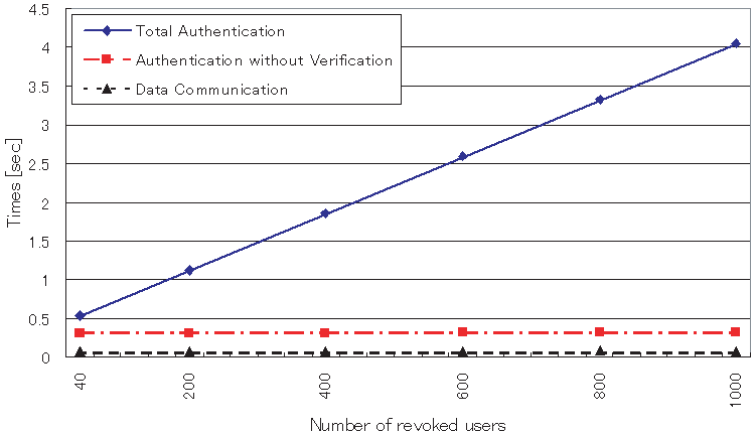

Fig. 5 Times of total authentication, authentication without verification and data communication.

number of the group members.

6.2 Performance Measurement

Figure 5 shows the total authentication time, the processing time without verification, and the communication time. The processing time without verification is the processing time excluding the verification on the server as the part of the authentication time that can be influenced by the specification of the mobile host. The communication time is the time of communication in STEP10 and STEP11. This time is given by measuring the time at the authentication server between the transmission of a 128-bit random number and the reception of the group signature from the mobile host, excluding the computational time of signing.

The total authentication time is about 0.54 seconds for $R = 40$, and 4.06 seconds for $R = 1{,}000$. On the other hand, the authentication time excluding the verification is about 0.3 seconds constantly. The packet size for communications is constantly about 572 bytes, and the communication only needs about 0.1 seconds constantly. This is because the packet size of the group signature is less than 1460 bytes, and the transmission of the group signature can be completed within only one frame, which has no overhead caused by packet divisions.

Finally, we show the processing time of **Open** algorithm to trace and identify the actual user. Essentially, this algorithm is divided into two phases, the signature verification and the decryption. The time of signature verification is about 37.8 ms and the time of the decryption is about 1.7 ms.

6.3 Discussion

The measurement results indicate that the load of the mobile host is very light, even if the specification of the mobile host is poor. This is the great advantage of our system using the VLR group signature scheme. Since the total authentication time grows linearly, the current implementation is suitable for middle-scale user groups such that R is less than thousands. The authentication times can be easily improved by using more powerful servers or enabling parallel computations of multi-core CPUs. Also, this can be improved by the more efficient implementation of pairings, which is currently a hot topic in cryptographic researches.

7. Conclusion

In this paper, we have presented an anonymous IEEE802.1X authentication system using a modified VLR group signature scheme. We implemented the main protocols of the system: authentication and user revocation protocols. The experimental results show that our system has the great advantage that the loads of mobile hosts are very light. Although the total authentication times depend on R, we have obtained practical times of a few seconds within $R = 1{,}000$.

Our future works include the reduction of the computational cost in the authentication server, and the implementations on other mobile devices such as PDAs and cellular phones.

Acknowledgments This work was partially supported by "R&D for advancement of functionality and usability in information history management" from the Ministry of Internal Affairs and Communications, Japan.

References

1) Cohen, H. and Frey, G.: Handbook of elliptic and hyperelliptic curve cryptography, Chapman & Hall/CRC (2005).
2) Wong, F.L., Lin, M., Nagaraja, S., Wassell, I. and Stajano, F.: Evaluation Framework of Location Privacy of Wireless Mobile Systems with Arbitrary Beam Pattern, *Proc. CNSR 2007*, pp.157–165 (2007).
3) Chaum, D. and Van Heijst, E.: Group signatures, *Proc. EUROCRYPT 1991*, LNCS 547, pp.241–246, Springer-Verlag (1991).
4) Tsudik, G. and Xu, S.: Accumulating composites and improved group signing, *Proc. ASIACRYPT 2003*, LNCS 2894, pp.269–286, Springer-Verlag (2003).
5) Camenisch, J. and Groth, J.: Group signature: Better efficiency and new theoretical aspects, *Proc. SCN 2004*, LNCS 3352, pp.120–133, Springer-Verlag (2004).
6) Camenisch, J. and Michels, M.: A group signature scheme based on an RSA-variant, *Proc. ASIACRYPT 1998*, LNCS 1514, pp.160–174, Springer-Verlag (1998).
7) Nakanishi, T., Hamada, N., Nakayama, T. and Funabiki, N.: Group signature schemes with efficient membership revocation using small primes, *Proc. WISA 2007*, pp.411–426 (2007).
8) Camenisch, J. and Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials, *Proc. CRYPTO 2002*, LNCS 2442, pp.61–67, Springer-Verlag (2002).
9) Nakanishi, T. and Funabiki, N.: Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps, *Proc. ASIACRYPT 2005*, LNCS 3788, pp.533–548, Springer-Verlag (2005).
10) Nakanishi, T. and Funabiki, N.: A short verifier-local revocation group signature scheme with backward unlinkability, *Proc. IWSEC 2006*, LNCS 4266, pp.17–32, Springer-Verlag (2006).
11) Nakanishi, T. and Funabiki, N.: Short verifier-local revocation group signature scheme with backward unlinkability, *IEICE Trans. Fundamentals*, Vol.E90-A, No.9, pp.1793–1802 (2007).
12) Boneh, D., Boyen, X. and Shacham, H.: Short group signatures, *Proc. Crypto 2004*, LNCS 3152, pp.41–55 (2004).
13) Boneh, D. and Shacham, H.: Group Signatures with Verifier-Local Revocation, *Proc. ACM-CCS 2004*, pp.168–177 (2004).
14) Isshiki, T., Mori, K., Sako, K., Teranishi, I. and Yonezawa, S.: Using group signatures for identity management and its implementation, *Proc. ACM-DIM 2006*, pp.73–78 (2006).
15) Furukawa, J. and Imai, H.: An efficient group signature scheme from bilinear maps, *Proc. ACISP 2005*, LNCS 3574, pp.455–467, Springer-Verlag (2005).
16) Barreto, P.S.L.M. and Naehrig, M.: Pairing-friendly elliptic curves of prime order, *Proc. SAC 2005*, LNCS 3897, pp.319–331 (2006).
17) Kato, H., Nogami, Y., Yoshida, T. and Morikawa, Y.: A multiplication algorithm in Fpm such that $p > m$ with a special class of gauss period normal bases, *IEICE Trans. Fundamentals*, Vol.E92-A, No.1, pp.173–181 (2009).
18) Akane, M., Nogami, Y. and Morikawa, Y.: Fast ate pairing computation of embedding degree 12 using subfield-twisted elliptic curve, *IEICE Trans. Fundamentals*, Vol.E92-A, No.2, pp.508–516 (2009).
19) Nogami, Y., Sakemi, Y., Okimoto, T., Nekado, K., Akane, M. and Morikawa, Y.:

Scalar multiplication using frobenius expansion over twisted elliptic curve for ate pairing based cryptography, *IEICE Trans. Fundamentals*, Vol.E92-A, No.1, pp.182–189 (2009).
20) Nogami, Y., Akane, M., Sakemi, Y., Kato, H. and Morikawa, Y.: Integer variable chi-based ate pairing, *Proc. Pairing 2008*, LNCS 5209, pp.178–191 (2008).
21) Sakemi, Y., Nogami, Y., Okeya, K., Kato, H. and Morikawa, Y.: Skew frobenius map and efficient scalar multiplication for pairing based cryptography, *Proc. CANS 2008*, LNCS 5339, pp.226–239 (2008).
22) Galbraith, S., Paterson, K. and Smart, N.: Pairing for cryptographers, *Discrete Applied Mathematics*, Vol.156, pp.3113–3121 (2008).
23) Chatterjee, S., Hankerson, D., Knapp, E. and Menezes, A.: Comparing two pairing-based aggregate signature schemes, preprint. http://eprint.iacr.org/2009/060
24) Congdon, P., Aboba, B., Smith, A., Zorn, G. and Roese, J.: IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines, IETF RFC 3580, (Sep. 2003).
25) Rigney, C., Willens, S., Rubens, A. and Simpson, W.: Remote authentication dial in user service, RFC 2865 (2000).
26) Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. and Levkowetz, H.: Extensible authentication protocol (EAP), IETF RFC 3748 (June 2004).
27) Aboba, B. and Simon, D.: PPP EAP TLS authentication protocol, IETF RFC 2716 (Oct. 1999).
28) Dierks, T. and Allen, C.: The TLS protocol, version 1.0, IETF RFC 2246 (Jan. 1999).
29) Funk, P. and Blake-Wilson, S.: EAP tunneled TLS authenticated protocol (EAP-TTLS), IETF RFC 5281 (Aug. 2008).
30) GNU multiple precision arithmetic library (GMP). http://gmplib.org/
31) Unison [Online]. http://www.cis.upenn.edu/~bcpierce/unison/

Appendix

A.1 Formal Security of Modified Scheme

A.1.1 Definitions

We discuss only the anonymity, since our modification has an influence on only the anonymity (the other requirements can be proved in the same way to the original). In the VLR group signature scheme, we should define the anonymity as the *BU-anonymity*, as well as Ref. 10). The BU-anonymity is the anonymity with the backward unlinkability. The backward unlinkability means that even after a revocation of a member, the signatures produced by the member before the revocation still remain anonymous.

The following is the version with the join protocol for the exculpability. Consider the following anonymity game.

Setup: The challenger runs **Setup**. He provides $\mathcal{A}$ with the public key, and runs $\mathcal{A}$. He sets $j = 0$, RU and CU with $\emptyset$.

Queries: $\mathcal{A}$ can query the challenger as follows.

- **H-Join:** $\mathcal{A}$ can request the i-th user's join. Then, the challenger executes the join protocol, where the challenger plays the both role of the joining user and GM.
- **C-Join:** $\mathcal{A}$ can request the i-th user's join. Then, $\mathcal{A}$ as the joining user executes the join protocol with the challenger as GM. The challenger adds i to CU.
- **Revocation:** $\mathcal{A}$ requests the revocation of a member i. The challenger increases j by 1, adds i to RU, and responds the revocation tokens for all members of RU at interval j.
- **Signing:** $\mathcal{A}$ requests a signature on a message M for a member i. The challenger responds the corresponding signature at j, if $i \notin CU$.
- **Corruption:** $\mathcal{A}$ requests the secret key of a member i. The challenger responds the secret key if $i \notin CU$. The challenger adds i to CU.
- **Opening:** $\mathcal{A}$ requests opening of a signature σ on a message M. The challenger responds the ID i of the signer, if σ is valid.

Challenge: $\mathcal{A}$ outputs a message M and two members i_0 and i_1. If $i_0 \notin CU$ and $i_1 \notin CU$, the challenger selects $\phi \in_R \{0, 1\}$, and responds the signature on M of member i_ϕ at the current $j = j^*$.

Restricted queries: Similarly, $\mathcal{A}$ can make the above queries. However, $\mathcal{A}$ cannot query the corruptions of i_0 and i_1, the revocations of i_0 and i_1 at the interval j^*, and opening of the challenged signature.

Output: Finally, $\mathcal{A}$ outputs a bit ϕ' indicating its guess of ϕ.

If $\phi' = \phi$, $\mathcal{A}$ wins. We define the advantage of $\mathcal{A}$ as $|\Pr[\phi' = \phi] - 1/2|$.

BU-anonymity requires that for all PPT $\mathcal{A}$, the advantage of $\mathcal{A}$ on this game is negligible.

A.1.2 Proof of Anonymity

We adopt the methodology "sequence of games." Consider the following sequence of games.

Game 0. This is the BU-anonymity game for the modified scheme. To response for the challenge query, note that $T_2 = \hat{f}^{\beta+y_i}$, and $T_3 = \hat{h}_j^{\beta}$ for $\beta \in_R Z_p$, where y_i is from $msk[i]$.

Game 1. The response for the challenge query in Game 0 is modified such that T_2 is randomly chosen from $\mathcal{G}_1$. The others are the same as Game 0.

Let S_0, S_1 denote the events that $\chi' = \chi$ in Game $0, 1$ respectively. In Game 1, the signature replied in the challenge query includes no information on $msk[i]$ except the ciphertext part (U, V, W), since the distributions of each value are uniformly random. This means that we can use the same proof as Ref. 15) to show that $|\Pr[S_1] - 1/2|$ is negligible under the DDH assumption.

Later, we will prove $|\Pr[S_0] - \Pr[S_1]|$ is negligible under the modified DLIN assumption. Then, $|\Pr[S_0] - 1/2|$ is also negligible, which means that the advantage of $\mathcal{A}$ is negligible, and thus the proposed scheme is BU-anonymous.

Hereafter we will construct an adversary $\mathcal{B}$ for the modified DLIN assumption using the adversary $\mathcal{A}$ for Game 0 and Game 1.

The input of $\mathcal{B}$ is $(u, v, w, \tilde{u}, \tilde{v}, \tilde{w}, u^a, v^b, Z)$, where $u \in_R \mathcal{G}_1$, $\tilde{u} \in_R \mathcal{G}_2$, $a, b, \rho_1, \rho_2 \in_R \mathbb{Z}_p^*$, $v = u^{\rho_1}, w = u^{\rho_2}, \tilde{v} = \tilde{u}^{\rho_1}, \tilde{w} = \tilde{u}^{\rho_2}$, and either $Z = w^{a+b}$ or $Z = w^c$ for $c \in_R \mathbb{Z}_p^*$. $\mathcal{B}$ decides which Z it is given by communicating with $\mathcal{A}$, as follows.

Setup: $\mathcal{B}$ picks $i^* \in_R [1, n]$ and $j^* \in_R [1, T]$. $\mathcal{B}$ simulates **Setup** as follows. $\mathcal{B}$ sets $g_1 = u$ and $g_2 = \tilde{u}$, and usually selects $\hat{g}_1, \tilde{g}_1 \in_R \mathcal{G}_1$. $\mathcal{B}$ selects $r_j \in_R \mathbb{Z}_p^*$ and computes $\hat{h}_j = g_1^{r_j}, h_j = g_2^{r_j}$ for all $j \in [1, T]$ except for j^*, and sets $\hat{h}_{j^*} = v, h_{j^*} = \tilde{v}$. Finally, $\mathcal{B}$ usually computes γ, Y, s, t, S, T.

Hash queries: At any time, $\mathcal{A}$ can query the hash function w.r.t. SPKs. $\mathcal{B}$ responds with random values with consistency.

Phase 1: $\mathcal{A}$ can request joining, signing, corruption, revocation, and opening queries at any time interval j. If $i \neq i^*$, then $\mathcal{B}$ responds to the query as usual. If $i = i^*$, $\mathcal{B}$ responds as follows.

H-Join: $\mathcal{B}$ sets $y_{i^*} = a$ which is unknown. Other values are generated as usual.

Sign: $\mathcal{B}$ computes a simulated group signature of i^* as follows.

1. $\mathcal{B}$ selects $r, \beta \in_R \mathbb{Z}_p^*$ and $T_1 \in_R \mathcal{G}_1$, and computes $\hat{f} = u^r$, $f = \tilde{u}^r$, $T_2 = (u^\beta u^a)^r = \hat{f}^{\beta+y_{i^*}}$, and $T_3 = \hat{h}_j^{\beta}$.
2. Compute U, V, W as usual, since x_{i^*} is known.
3. $\mathcal{B}$ computes the simulated SPK X by using the simulator of the perfect zero-knowledge-ness.

Then, $\mathcal{B}$ responds signature $\sigma = (T_1, T_2, T_3, \hat{f}, f, U, V, W, X)$ to $\mathcal{A}$.

Revocation: If $j \neq j^*$, $\mathcal{B}$ responds $B_{i^*j} = (u^a)^{r_j} = \hat{h}_j^{y_{i^*}}$. Otherwise, $\mathcal{B}$ outputs a random guess $\omega' \in_R \{0, 1\}$ and aborts.

Corruption: $\mathcal{B}$ outputs a random guess $\omega' \in_R \{0, 1\}$ and aborts.

Opening: $\mathcal{B}$ usually decrypts the ciphertext (U, V, W) in the queried signature, using s.

Challenge: $\mathcal{A}$ outputs a message M, the current time interval j and two members i_0 and i_1 to be challenged. If $j \neq j^*$, $\mathcal{B}$ outputs a random guess $\omega' \in_R \{0, 1\}$ and aborts. Otherwise, $\mathcal{B}$ picks $\phi \in_R \{0, 1\}$. Then, if $i_\phi \neq i^*$, $\mathcal{B}$ outputs a random guess $\omega' \in_R \{0, 1\}$ and aborts. Otherwise, $\mathcal{B}$ responds the following simulated group signature of i^* and j^*.

1. $\mathcal{B}$ selects $r \in_R \mathbb{Z}_p^*$ and sets $\beta = b$ (unknown). $\mathcal{B}$ selects $T_1 \in_R \mathcal{G}_1$, sets $\hat{f} = w$, $f = \tilde{w}$, $T_2 = Z$, and $T_3 = v^b = \hat{h}_{j^*}^{\beta}$. If $Z = w^{a+b}$, $T_2 = w^{a+b} = \hat{f}^{\beta+y_{i^*}}$. Otherwise, T_2 is a random from $\mathcal{G}_1$.
2. $\mathcal{B}$ computes the simulated SPK X similarly.

Phase 2: This is the same as **Phase 1**.

Output: $\mathcal{A}$ outputs its guess $\phi \in \{0, 1\}$. If $\phi = \phi'$, $\mathcal{B}$ outputs $\omega' = 1$ (implying $Z = w^{a+b}$), and otherwise outputs $\omega' = 0$ (implying $Z = w^c$).

$\varepsilon_{\mathrm{mDLIN}}$ denotes the advantage of $\mathcal{B}$. As well as Ref. 10), we can evaluate the advantage $\varepsilon_{\mathrm{mDLIN}}$ by using $\Pr[S_0]$ and $\Pr[S_1]$, and obtain $|\Pr[S_0] - \Pr[S_1]| \leq nT\varepsilon_{\mathrm{mDLIN}}$. Since $\varepsilon_{\mathrm{mDLIN}}$ is negligible under the modified DLIN assumption, and since n and T are polynomially bounded, $|\Pr[S_0] - \Pr[S_1]|$ is also negligible.

(Received May 18, 2009)
(Accepted December 17, 2009)
(Released March 10, 2010)

Amang Sudarsono received his B.E. degree in electrical engineering, telecommunication and multimedia program from Sepuluh Nopember Institute of Technology, Indonesia, in 2001. From 1997 to 2002, he was with the Network Engineering Division, Metro Cellular Nusantara, Ltd., Indonesia. He joined the Department of Telecommunication Technology at Electronics Engineering Polytechnic Institute of Surabaya (EEPIS), Indonesia, as a lecturer in 2002. Currently, he is a Ph.D. candidate in Graduate School of Natural Science and Technology at Okayama University, Japan. His research interests include group signatures and network securities.

Toru Nakanishi received his M.S. and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1995 and 2000, respectively. He joined the Department of Information Technology at Okayama University, Japan, as a research associate in 1998, and moved to the Department of Communication Network Engineering in 2000, where he became an assistant professor and an associate professor in 2003 and 2006, respectively. His research interests include cryptography and information security. He is a member of the IEICE.

Yasuyuki Nogami received his B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Shinshu University in 1994, 1996, and 1999, respectively. He is currently a research associate in the Department of Communication Network Engineering at Okayama University. His research interests include the finite field theory and its applications. He is a member of the IEICE and IEEE.

Nobuo Funabiki received his B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received his M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000–2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering at Okayama University as a professor. His research interests include network protocols, optimization algorithms, image processing, educational technology, and network security. Dr. Funabiki is a member of IEICE and IEEE.