An Efficient Group Signature Scheme from Bilinear Maps

Jun Furukawa[1,2] and Hideki Imai[2]

[1] NEC Corporation, 1753, Shimonumabe, Nakahara, Kawasaki 211-8666, Japan
j-furukawa@ay.jp.nec.com
[2] Institute of Industrial Science, The University of Tokyo, 4-6-1, Komaba, Meguro, Tokyo 153-8505, Japan
imai@iis.u-tokyo.ac.jp

Abstract. We propose a new group signature scheme which is secure if we assume the Decision Diffie-Hellman assumption, the q-Strong Diffie-Hellman assumption, and the existence of random oracles. The proposed scheme is the most efficient among the all previous group signature schemes in signature length and in computational complexity.

1 Introduction

A group signature scheme, first proposed by Chaum and van Heyst [13] and followed by [1,2,6,8,10,11,12,27], allows each member of a group to sign messages on behalf of the group without revealing his own identity. The scheme also realizes a special authority that can identify actual signers in case of dispute. Group signatures have many applications in which user anonymity is required such as in anonymous credential systems [2], identity escrow [21,20], voting and bidding [1], and electronic cash systems.

Although earlier group signature schemes required large computational cost and long signatures, recently proposed schemes, such as the one proposed by Ateniese et al. in [1], are very efficient. In particular, Boneh, Boyen, and Shacham [7], Nguyen and Safavi-Naini [27], and Camenisch and Lysyanskaya [10] proposed very efficient group signature schemes based on bilinear maps. Currently, the most efficient construction is the one proposed in [7]. The signature length of the scheme in [7] is 42% and 38% of those of [27] and [10] respectively. The computational cost for the scheme in [7] is also smaller than those of [27] and [10][3].

This paper proposes a novel group signature scheme based on bilinear maps. Our scheme is more efficient than any of the previous schemes. Moreover, our scheme requires fewer assumptions than the scheme in [7], which is the most efficient among the previous schemes.

[3] The heaviest computation in these schemes is computation of a bilinear map such as Tate pairing. As shown in Table 10 in [17], its computational cost is smaller than that of computation of full-exponent RSA.

C. Boyd and J.M. González Nieto (Eds.): ACISP 2005, LNCS 3574, pp. 455–467, 2005.

Our approach to the construction of a group signature scheme is similar to that adopted by Boneh et al. in [7]. They used a set of three groups $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_T$ of the same prime order p such that there exists a bilinear map from $\mathcal{G}_1 \times \mathcal{G}_2$ to $\mathcal{G}_T$. Each group member has a pair comprising a membership certificate and a membership secret with which he signs on behalf of the group. The membership certificate and membership secret are elements of $\mathcal{G}_1$ and $\mathbb{Z}/p\mathbb{Z}$. For a special authority to identify actual signers from group signatures in their scheme, signers are required to attach an encryption of a part of the membership certificate which is an element of $\mathcal{G}_1$. Because of the existence of the bilinear map, their scheme is not able to simply use ElGamal encryption scheme for this purpose. Hence, they introduced a new encryption scheme called "linear encryption scheme" based on a new assumption called the Decision Linear Diffie-Hellman (DLDH) assumption. This encryption scheme is more complex than the ordinary ElGamal type encryption scheme.

The main difference between our approach and that in [7] is that we use a group $\mathcal{G}$ of the same order p in addition to the three groups $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_T$ such that the Decision Diffie-Hellman (DDH) problem on $\mathcal{G}$ is difficult to solve. For a special authority to identify actual signers from group signatures in our scheme, signers are required to attach an encryption of the exponentiation of the membership secret in $\mathcal{G}$. Because this exponentiation to be encrypted is in $\mathcal{G}$, we can apply a simple ElGamal type encryption scheme. This makes our scheme more efficient and requires fewer assumptions than the scheme in [7].

For the groups $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_T$ and their associated bilinear map, we can use, for example, the elliptic curve proposed by [26] (MNT curve) and Tate pairing. The choice of such a curve makes it possible to express elements in $\mathcal{G}_1$ by a short string. Although the number of such curves are found in [26] is small, more MNT curves are found in [30]. Therefore, since we can easily find an elliptic curve of the same given order p as $\mathcal{G}$ with practically high probability by using a complex multiplication method, finding a desired set of $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G})$ is practical.

As a result, our signature lengths are, respectively, 83%, 36%, and 32% of those of signatures in [7], [27], and [10] if we choose groups so that elements of $\mathcal{G}_1$, $\mathcal{G}_T$, and $\mathcal{G}$ can be expressed in 171, 1020, and 171 bit strings respectively. Although we cannot a present precise estimation of the computational cost since it depends on the choice of groups, our scheme requires less computational cost than any of the schemes in [7,27,10]. The security of our scheme depends on the DDH assumption, the Strong Diffie-Hellman (SDH) assumption, and the existence of random oracles. We do not present how to revoke group members. However, the revocation mechanisms described in [7] can be also applied to our system. In our scheme, group members are able to determine their secret key when they join the group, which enables them to join many groups using the same secret key. This property may reduce operational cost when there are many groups. The scheme in [27] does not have such a property. (The scheme in [10] does.)

Our paper is organized as follows. Section 2 describes the model and security requirements of the group signature scheme and notation and complexity

assumptions. Section 3 proposes our group signature scheme, and Section 4 discusses its security. Section 5 compares our scheme with the previous schemes.

2 Background

2.1 Model of Group Signature Scheme

Let $b \leftarrow \mathsf{AL}(a)$ denote an algorithm AL, where its input is a and its output is b. Let $\langle c, d\rangle \leftarrow \mathsf{IP}_{A,B}\langle a, b\rangle$ denote an interactive protocol IP between A and B, where private inputs to A and B are, respectively, a and b, and outputs of A and B are, respectively, c and d.

The model of the group signature scheme is defined as follows. In this model, we do not consider revocation for the sake of simplicity.

Definition 1. *Players in the group signature scheme are a membership manager MM, a tracing manager TM, a group member U and a verifier V. $k \in \mathbb{N}$ is a security parameter.*

A group signature scheme $\mathcal{GS}$ *consists of the following five algorithms and one interactive protocol.* ($\mathsf{M\text{-}KeyGen}$, $\mathsf{T\text{-}KeyGen}$, Join, Sign, Verify, Open),

- *A probabilistic key generation algorithm for MM that, given a security parameter 1^k, outputs a* membership public key *mpk and a* membership secret key *msk.*

$$(msk, mpk) \leftarrow \mathsf{M\text{-}KeyGen}(1^k)$$

- *A probabilistic key generation algorithm for TM that, given* mpk*, outputs a* tracing public key *tpk and a* tracing secret key *tsk.*

$$(tsk, tpk) \leftarrow \mathsf{T\text{-}KeyGen}(mpk)$$

- *An interactive member registration protocol for the MM and a user U. MM is given* mpk, msk*, the user's identity U*[4]*, and a list of all group members $\mathcal{L}$. U is given* mpk*. If the interaction was successful, U outputs a* membership certificate $cert_U$*, a* membership secret sk_U*, and an* identifier $ider_U$ *and MM adds a pair $(U, ider_U)$ to $\mathcal{L}$ and outputs this revised $\mathcal{L}$.*

$$\langle(\mathcal{L}), (cert_U, sk_U, ider_U)\rangle \leftarrow \mathsf{Join}_{MM,U}\langle(\mathcal{L}, U, mpk, msk), (mpk)\rangle$$

- *A probabilistic signature generation algorithm for a U that, given* mpk*, tpk, $cert_U, sk_U$, and a* message m*, outputs a group signature gs on the message m.*

$$gs \leftarrow \mathsf{Sign}(mpk, tpk, cert_U, sk_U, m)$$

[4] We use the same notation U for a user and the identity of this user U.

– *A deterministic signature verification algorithm for any V that, given* mpk, *tpk, m, and gs, returns either* acc *or* rej. *Here,* acc *and* rej *represent, respectively, an acceptance and a rejection of the signature.*

$$acc/rej \leftarrow \mathsf{Verify}(mpk, tpk, m, gs)$$

We say that a group signature gs on m is valid *if* acc $\leftarrow \mathsf{Verify}(mpk, tpk, m, gs)$.

– *A deterministic signer tracing algorithm for the TM that, given* mpk, *tpk, tsk, m, and gs, outputs $\perp$ if gs on m is not valid. Otherwise, it outputs $(U, proof)$, where proof assures the validity of the result U. If the algorithm cannot find the actual signer in $\mathcal{L}$, the algorithm outputs $\perp'$ instead of U.*

$$\perp/(U/\perp', proof) \leftarrow \mathsf{Open}(mpk, tpk, tsk, m, gs, \mathcal{L})$$

2.2 Security Requirements

Security requirements for group signature schemes that includes a dynamically changing membership and separation of group manager into membership manager and tracing manager are proposed in [4,16,18]. In [4], Bellare et al. called these requirements Traceability, Anonymity, and Non-frameability. Requirements in [16,18] are basically the same.

Roughly, Traceability guarantees that no one except the MM is able to successfully add a new member to the group. Anonymity guarantees that no one except the TM is able to successfully identify actual signers of signatures. Non-Frameability guarantees that no one except each member is able to successfully create a signature which will be linked to his own identity when opened by the TM.

We give short description of these requirements with minor modifications, which do not consider revocation for the sake of simplicity.

Definition 2. *(***Traceability***) Let $\mathcal{GS}$ be a group signature scheme, and let $\mathcal{A}$ be an algorithm. We consider the following experiment that returns* 0/1. *Here, we assume that* Join *protocols are executed only sequentially.*

Experiment $\mathbf{Exp}^{\mathrm{Tr}}_{\mathcal{GS},\mathcal{A}}(k)$
  $(mpk, msk) \leftarrow \mathsf{M\text{-}KeyGen}(1^k)$
  $(tpk, State) \leftarrow \mathcal{A}(mpk)$
  $Cont \leftarrow \mathsf{true}$
  While $Cont = \mathsf{true}$ do
    $\langle(\mathcal{L}), (State)\rangle \leftarrow \mathsf{Join}_{MM,A}\langle(\mathcal{L}, U, mpk, msk), (mpk, State)\rangle$
  EndWhile
  $(m, gs) \leftarrow \mathcal{A}(State)$
  If $rej \leftarrow \mathsf{Verify}(mpk, tpk, m, gs)$ then return 0
  If $(\perp', proof) \leftarrow \mathsf{Open}(mpk, tpk, tsk, m, gs, \mathcal{L})$ then return 1
  Return 0

A group signature scheme $\mathcal{GS}$ has traceability *property if for all probabilistic, polynomial-time machines $\mathcal{A}$,*

$$\Pr[\mathbf{Exp}^{\mathrm{Tr}}_{\mathcal{GS},\mathcal{A}}(k) = 1]$$

is negligible in k.

Definition 3. *(***Anonymity***) Let $\mathcal{GS}$ be a group signature scheme, let $b \in \{0,1\}$, and let $\mathcal{A}$ be an algorithm. We consider the following experiment that returns* $0/1$.

Experiment $\mathbf{Exp}^{\mathrm{An}}_{\mathcal{GS},\mathcal{A}}(k,b)$
$(mpk, State) \leftarrow \mathcal{A}(1^k)$
$(tpk, tsk) \leftarrow \mathsf{T\text{-}KeyGen}(mpk)$
$(State, (cert_0, sk_0), (cert_1, sk_1), m) \leftarrow \mathcal{A}^{\mathsf{Open}(mpk,tpk,tsk,\cdot,\cdot,\cdot)}(State, tpk)$
$gs \leftarrow \mathsf{Sign}(mpk, tpk, cert_b, sk_b, m)$
$(b' \in \{0,1\} \leftarrow \mathcal{A}^{\mathsf{Open}(mpk,tpk,tsk,\cdot,\cdot,\cdot)}(State, gs)$
If $\mathcal{A}$ did not query Open oracle with (m, gs) after gs is given, then return b'
Return 0

A group signature scheme $\mathcal{GS}$ has anonymity *property if for all probabilistic polynomial-time machines $\mathcal{A}$,*

$$\Pr[\mathbf{Exp}^{\mathrm{An}}_{\mathcal{GS},\mathcal{A}}(k,0) = 1] - \Pr[\mathbf{Exp}^{\mathrm{An}}_{\mathcal{GS},\mathcal{A}}(k,1) = 1]$$

is negligible in k.

Definition 4. *(***Non-Frameability***) Let $\mathcal{GS}$ be a group signature scheme, and let $\mathcal{A}$ be an algorithm. We consider the following experiment that returns* $0/1$.

Experiment $\mathbf{Exp}^{\mathrm{NF}}_{\mathcal{GS},\mathcal{A}}(k)$
$(mpk, tpk, State) \leftarrow \mathcal{A}(1^k)$
$\langle State, (cert_U, sk_U, ider_U)\rangle \leftarrow \mathsf{Join}_{A,U}\langle State, mpk\rangle$
If the tuple $(cert_U, sk_U, ider_U)$ is not valid then return 0
$(m, gs, \mathcal{L}) \leftarrow \mathcal{A}^{\mathsf{Sign}(mpk,tpk,cert_U,sk_U,\cdot)}(State)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(U, ider_U)\}$
If $rej \leftarrow \mathsf{Verify}(mpk, tpk, m, gs)0$ then return 0
If $(U, proof) \leftarrow \mathsf{Open}(mpk, tpk, tsk, m, gs, \mathcal{L})$ and m was not queried by $\mathcal{A}$ to the signing oracle Sign then return 1
Else return 0

A group signature scheme $\mathcal{GS}$ has Non-frameability *property if for all probabilistic polynomial-time machines $\mathcal{A}$,*

$$\Pr[\mathbf{Exp}^{\mathrm{NF}}_{\mathcal{GS},\mathcal{A}}(k) = 1]$$

is negligible in k.

2.3 Notation and Complexity Assumption

Let $\mathcal{G}_{1k}, \mathcal{G}_{2k}$, and $\mathcal{G}_k$ be a cyclic group of length k prime order p. We omit index k if not confusing. Let G_1, G_2, and G be, respectively, generators of $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}$. Let ψ be an isomorphism from $\mathcal{G}_2$ to $\mathcal{G}_1$, with $\psi(G_2) = G_1$. Let e be a bilinear map $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$. Let $\mathcal{H}$ be a hash function that maps string to $\mathbb{Z}/p\mathbb{Z}$.

Definition 5. (Decision Diffie-Hellman assumption) *Let the Decision Diffie-Hellman problem in $\mathcal{G}_k$ be defined as follows: given 4-tuple $(G, [a]G, [b]G, [c]G) \in (\mathcal{G}_k)^4$ as input, output 1 if $c = ab$ and 0 otherwise. An algorithm $\mathcal{A}$ has advantage $\epsilon(k)$ in solving the Decision Diffie-Hellman problem in $\mathcal{G}_k$ if*

$$|\Pr[A(G, [a]G, [b]G, [ab]G) = 1] - \Pr[A(G, [a]G, [b]G, [c]G) = 1]| \geq \epsilon(k)$$

where the probability is taken over the random choice of generator G in $\mathcal{G}_k$, of $(a, b, c) \in (\mathbb{Z}/p\mathbb{Z})^3$, and of the random tape of $\mathcal{A}$.

We say that the Decision Diffie-Hellman assumption holds in $\{\mathcal{G}_k\}_{k\in\mathbb{N}}$ if no polynomial-time algorithm has advantage $\epsilon(k)$ non-negligible in k in solving the Decision Diffie-Hellman problem in $\mathcal{G}_k$.

Definition 6. (Strong Diffie-Hellman Assumption) *Let the q-Strong Diffie-Hellman Problem (q-SDH) in $(\mathcal{G}_{1k}, \mathcal{G}_{2k})$ be defined as follows: given a $(q+2)$-tuple $(G_1, G_2, [\gamma]G_2, [\gamma^2]G_2, \ldots, [\gamma^q]G_2) \in \mathcal{G}_{1k} \times (\mathcal{G}_{2k})^{q+1}$ as input, output a pair $([1/(x+\gamma)]G_1, x)$ where $x \in \mathbb{Z}/p\mathbb{Z}$. An algorithm $\mathcal{A}$ has advantage $\epsilon(k)$ in solving the q-SDH problem in $(\mathcal{G}_{1k}, \mathcal{G}_{2k})$ if*

$$\Pr[A(G_1, G_2, [\gamma]G_2, \ldots, [\gamma^q]G_2) = ([1/(x+\gamma)]G_1, x)] \geq \epsilon(k),$$

where the probability is taken over the random choice of generator G_2 in $\mathcal{G}_{2k}$ (with $G_1 = \psi(G_2)$), of $\gamma \in \mathbb{Z}/p\mathbb{Z}$, and of the random tape of $\mathcal{A}$.

We say that the Strong Diffie-Hellman (SDH) assumption holds in $\{(\mathcal{G}_{1k}, \mathcal{G}_{2k})\}_{k\in\mathbb{N}}$ if no polynomial-time algorithm has advantage $\epsilon(k)$ non-negligible in k in solving the q-SDH problem in $(\mathcal{G}_{1k}, \mathcal{G}_{2k})$ for q polynomial of k.

The SDH assumption is proposed and proved to hold in generic bilinear groups in [6]. This assumption is a variant of an assumption proposed by Mitsunari et al. in [25].

3 Proposed Group Signature Scheme

Now we will present our efficient group signature scheme.

M-KeyGen

Given 1^k, M-KeyGen chooses $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T$ such that its order p is of length k and then randomly chooses $w \in_R \mathbb{Z}/p\mathbb{Z}$ and $(H, K) \in_R (\mathcal{G}_1)^2$ and generates $Y = [w]G_2$. Then, M-KeyGen outputs

$$(msk, mpk) := (w, (p, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, e, \mathcal{G}, G_1, G_2, G, \psi, \mathcal{H}, Y, H, K))$$

Here, some of the symbols are interpreted as binary strings that describe those symbols. For example, $\mathcal{G}$ expresses the string of the document that specifies group $\mathcal{G}$.

T-KeyGen

Given mpk, T-KeyGen first randomly chooses $(s,t) \in_R (\mathbb{Z}/p\mathbb{Z})^2$. Next, T-KeyGen generates $(S,T) = ([s]G, [t]G)$. Finally, T-KeyGen outputs

$$(tsk, tpk) := ((s,t),(S,T)) .$$

Join$_{MM,U}$

1. – MM is given group member list $\mathcal{L}$, an identity of a user U, mpk, and msk.
   – A user U is given mpk.
2. U randomly chooses $sk_U := x_U \in_R \mathbb{Z}/p\mathbb{Z}$ and $z'_U \in_R \mathbb{Z}/p\mathbb{Z}$ and generates

$$ider_U := Q_U = [x_U]G \ , \ H_U = [x_U]H + [z'_U]K$$

and sends $(Q_U, H_U)$to MM[5].
Then, U proves in zero-knowledge to MM the knowledge of x_U and z'_U as follows. Although the protocol given here is only honest verifier zero-knowledge, from this we can construct a black-box zero-knowledge protocol using the technique presented in [24]. We still assume that Join protocols are executed in a sequential manner (or concurrently but with an appropriate timing-constraint [14]).

(i) U randomly chooses $(x'_U, z') \in_R (\mathbb{Z}/p\mathbb{Z})^2$ and generates

$$Q'_U = [x'_U]G \ , \ H'_U = [x'_U]H + [z']K$$

and sends them to MM.

(ii) MM sends U randomly chosen $c_U \in_R \mathbb{Z}/p\mathbb{Z}$.

(iii) U generates

$$r_U = c_U x_U + x'_U \ , \ s_U = c_U z'_U + z'$$

and sends (r_U, s_U) to MM.

(iv) MM checks that the following equations hold:

$$[r_U]G = [c_U]Q_U + Q'_U \ , \ [r_U]H + [s_U]K = [c_U]H_U + H'_U$$

3. The MM randomly chooses $(y_U, z''_U) \in_R (\mathbb{Z}/p\mathbb{Z})^2$ and generates

$$A_U = [1/(w + y_U)](G_1 - H_U - [z''_U]K)$$

and sends (A_U, y_U, z''_U) to U. The MM adds an entry $(U, ider_U) = (U, Q_U)$ to its group member list $\mathcal{L}$.

[5] U needs to sign on Q_U to prove that U agreed to be a group member; we omit this process for the sake of simplicity.

4. U generates its membership certificate as

$$cert_U := (A_U, y_U, z_U) = (A_U, y_U, z'_U + z''_U).$$

U checks that the following equation holds:

$$e(A_U, Y + [y_U]G_2) \cdot e([x_U]H, G_2) \cdot e([z_U]K, G_2) = e(G_1, G_2).$$

5. – MM outputs the revised $\mathcal{L}$.
   – U outputs $(cert_U, sk_U, ider_U) = ((A_U, y_U, z_U), x_U, Q_U)$.

Remark 1. Publishing $(cert_U, ider_U)$ which MM is able to obtain does not compromise the security of the system.

Sign

1. Sign is given $mpk, tpk, cert_U, sk_U$, and m.
2. Sign randomly chooses $(r, q) \in_R (\mathbb{Z}/p\mathbb{Z})^2$ and generates

$$B = A_U + [q]K \ , \quad U = [x_U + r]G \ , \quad V = [r]S \ , \quad W = [r]T \tag{1}$$

Here, the following equation holds.

$$\begin{aligned} &e(G_1, G_2) \\ &= e(B, Y) \cdot e(H, G_2)^{x_U} \cdot e(B, G_2)^{y_U} \cdot e(K, G_2)^{z_U - q\, y_U} \cdot e(K, Y)^{-q} \end{aligned} \tag{2}$$

The data generated hereafter is a Fiat-Shamir transformation of a zero-knowledge proof of knowledge of x_U, y_U, z_U, and q, r that satisfies Eqs. (1) and (2). Since B is a perfect hiding commitment of A_U, the only knowledge that the receiver of the signature can obtain is (U, V, W) which is an ElGamal type double encryption of $[x_U]G$

(i) Sign randomly chooses $(t, u, v, f, o) \in_R (\mathbb{Z}/p\mathbb{Z})^5$ and generates

$$\begin{aligned} X' &= e(H, G_2)^t \cdot e(B, G_2)^u \cdot e(K, G_2)^v \cdot e(K, Y)^f \\ U' &= [t + o]G \ , \quad V' = [o]S \ , \quad W' = [o]T \end{aligned}$$

(ii) Sign generates

$$c = \mathcal{H}(p, G_1, G_2, G_T, G, \psi, Y, S, T, H, K, B, U, V, W, X', V', W', U', m)$$

(iii) Sign generates

$$\begin{aligned} x' &= cx_U + t \ , \quad y' = cy_U + u \ , \quad z' = c(z_U - qy_U) + v \\ q' &= -cq + f \ , \quad r' = cr + o \end{aligned}$$

3. Sign outputs

$$gs := (B, U, V, W, c, x', y', z', q', r')$$

as a signature on message m.

Verify

1. Verify is given mpk, tpk, m, and gs.
2. Verify generates

$$X' = e(H, G_2)^{x'} e(B, G_2)^{y'} e(K, G_2)^{z'} e(K, Y)^{q'} \left(\frac{e(G_1, G_2)}{e(B, Y)} \right)^{-c}$$
$$U' = [x' + r']G - [c]U \ , \quad V' = [r']S - [c]V \ , \quad W' = [r']T - [c]W.$$

3. Verify outputs acc if equation

$$c = \mathcal{H}(p, G_1, G_2, G_T, G, \psi, Y, S, T, H, K, B, U, V, W, X', V', W', U', m)$$

holds. Otherwise, it outputs rej.

Open

1. Open is given mpk, tpk, tsk, m, gs, and $\mathcal{L}$.
2. If $\mathsf{Verify}(mpk, tpk, m, gs) = rej$, it outputs $\perp$ and stops.
3. Open generates and outputs

$$Q = U - [1/s]V \qquad (= U - [1/t]W)$$

Then, Open generates and outputs a non-interactive proof of knowledge of either s or t that satisfies either of the above equations and Q as a $proof$.
4. Open searches Q_U that coincides with the Q in $\mathcal{L}$. If there is such a Q_U, it outputs the corresponding U. Otherwise, it outputs $\perp'$.

4 Security

Theorem 1. *The proposed scheme has Traceability property if the SDH assumption holds.*

Theorem 2. *The proposed scheme has Anonymity property if the DDH assumption holds.*

Theorem 3. *The proposed scheme has Non-Frameability property if we assume the discrete logarithm problem is difficult to solve.*

Proofs of the theorems are given in the full paper [15].

5 Comparison with Previous Schemes

We compare the signature length and computational complexity of the proposed scheme to those of the previous schemes [27,10] and those of a variant of the scheme in [7]. This variant protocol is given in the full paper [15].

The variant scheme of [7] differs from the original one in two points. The first point is that it provides a joining protocol, whose construction is already presented in Section 7 of [6]. The second point is that it uses a double encryption scheme [28] variant of the linear encryption scheme instead of the simple linear encryption scheme used in the original scheme. Since the *Open* oracle in group signature plays a role similar to that of the role of the decryption oracle in the IND-CCA2 game of public key cryptosystems, the encryption scheme used in group signature needs to be IND-CCA2 secure. However, the signed ElGamal encryption is IND-CCA2 secure only in the generic model [31], in the same way that the linear encryption scheme adopted in [7] is. Hence, the use of a double encryption variant is a legitimate solution to avoid dependence on the generic group model.

Although the above variant scheme is less efficient than the original scheme, comparing our scheme with this variant scheme is appropriate. This is because our scheme and the schemes in [27] and [10] all provide a Join protocol and their security is proved in a non-generic group model.

We compare the group signature lengths of our scheme and those of the previous schemes. We assume that $\mathcal{G}_1 \neq \mathcal{G}_2$ such that the representation of G_1 can be a 172 bit string when $|p| = 171$ by using the elliptic curve defined by [26]. The choice of such a curve makes it possible to express B by a short string. When such a curve is not available, the signature length of our scheme is much shorter than those of the other previous schemes. We also assume that the representations of G_T and G are 1020 bits and 172 bits. A group signature of the variant of the scheme in [7] is composed of seven $\mathbb{Z}/p\mathbb{Z}$ and five $\mathcal{G}_1$ elements. That of the scheme in [27] is composed of ten $\mathbb{Z}/p\mathbb{Z}$, six $\mathcal{G}_1$, and two $\mathcal{G}_T$ elements, and that of the scheme in [10] is composed of four $\mathbb{Z}/p\mathbb{Z}$, three $\mathcal{G}_1$, and four $\mathcal{G}_T$ elements. In contrast, that of the proposed scheme is composed of six $\mathbb{Z}/p\mathbb{Z}$, one $\mathcal{G}_1$, and three $\mathcal{G}$ elements, and thus its signature length is the shortest among the other previous schemes.

We also estimate the computational cost of our scheme and that of the previous schemes by the number of scalar multiplications/modular exponentiations in $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_T$ and the number of pairing operations e required for Sign and Verify, since these are the most costly computations. Although we cannot present a precise estimation of the computational cost of each operation since it depends on the choice of the groups $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_T$, these computations can be done quite efficiently if we choose Tate pairing for e and adopt the computation tools described in [23].

We also list the assumptions required in our scheme and the previous schemes [27,10], and the variant of the scheme in [7]. From Theorems 1, 2, and 3, our scheme requires the SDH assumption, the Decision Diffie-Hellman assumption, and the existence of random oracles. The scheme in [7] requires the SDH as-

sumption, the DLDH assumption, and the existence of random oracles. That in [27] requires the SDH assumption, the Decision Bilinear Diffie-Hellman (DBDH) assumption, and the existence of random oracles. That in [10] requires the Lysyanskaya-Rivest-Sahai-Wolf (LRSW) assumption, the Decision Diffie-Hellman assumption, and the existence of random oracles. The DLDH assumption is proposed in [7] which is proved to hold in generic bilinear groups. The LRSW assumption is proposed in [22] and is proved to hold in generic groups. The LRSW assumption is also proved to hold in generic bilinear groups in [10]. The SDH assumption and the LRSW assumption cannot be compares to each other.

These results of estimation and required assumptions are given in Table 1, where "# of SMul" , "# of MExp", "# of pairings", and "Sig. Len." are abbreviations of "the number of scalar multiplications" , "the number of modular exponentiations", "the number of pairings", and "signature length". Installing the revocation mechanism proposed in [7] has no effect on this estimation[6].

	A variant of [7] Sign/Verify	Scheme in [27] Sign/Verify	Scheme in [10] Sign/Verify	Our Scheme Sign/Verify
# of SMul in $\mathcal{G}$	-	-	-	7/6
# of SMul in $\mathcal{G}_2$	13/12	20/13	3/0	-
# of MExp in $\mathcal{G}_T$	4/5	6/2	14/16	4/5
# of pairings	0/2	0/3	0/3	0/2
Sig. Len. (bits)	2057	4782	5296	1711
Assumptions	SDH,DLDH	SDH,DBDH	LRSW,DDH	SDH,DDH

Table 1. Complexity & Assumptions

References

1. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik: A Practical and Provable Secure Coalition-Resistant Group Signature Scheme. CRYPTO 2000, LNCS 1880, pp.255–270.
2. G. Ateniese, B. de Medeiros: Efficient Group Signatures without Trapdoors. ASIACRYPT 2003, LNCS 2894, pp.246–268.
3. Niko Bari, Birgit Pfitzmann: Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. EUROCRYPT 1997: 480-494.
4. Mihir Bellare, Haixia Shi, Chong Zhang: Foundations of Group Signatures: The Case of Dynamic Groups. CT-RSA 2005: 136-153
5. M. Bellare, D. Micciancio, B. Warinschi: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. EUROCRYPT 2003, LNCS 2656, pp. 614-629.

[6] Given $(G_1, G_2, A_U, y_U, x_U, z_U, y_{\bar{U}}, \bar{H}, \bar{K}, Y)$ such that $Y = [w]G_2, [w + y_U]A_U + [x_U]H + [z_U]K = G_1, [w + y_{\bar{U}}]\bar{G}_1 = G_1, [w + y_{\bar{U}}]\bar{H} = H, [w + y_{\bar{U}}]\bar{K} = K$ for some $w \in \mathbb{Z}/q\mathbb{Z}$, $\bar{A}_U$ that satisfies $[w + y_U]\bar{A}_U + [x_U]\bar{H} + [z_U]\bar{K} = \bar{G}_1$ can be computed as $\bar{A}_U = [1/(y_{\bar{U}} - y_U)](A_U - \bar{G}_1 - [x_U]\bar{H} - [z_U]\bar{K})$.

6. Dan Boneh, Xavier Boyen: Short Signatures Without Random Oracles. EUROCRYPT 2004: 56-73.
7. Dan Boneh, Xavier Boyen, Hovav Shacham: Short Group Signature. CRYPTO 2004, Lecture Notes in Computer Science 3152, pp. 41-55, 2004, Springer.
8. Jan Camenisch, Jens Groth: Group Signatures: Better Efficiency and New Theoretical Aspects. Security in Communication Networks - SCN 2004, LNCS series.
9. Jan Camenisch, Anna Lysyanskaya: A Signature Scheme with Efficient Protocols. SCN 2002: 268-289.
10. Jan Camenisch, Anna Lysyanskaya: Signature Schemes and Anonymous Credentials from Bilinear Maps. Crypto 2004, Springer Verlag, 2004.
11. J. Camenisch, M. Michels: A group signature scheme based on an RSA-variant. Technical Report RS-98-27, BRICS, University of Aarhus, November 1998. An earlier version appears in ASIACRYPT '98.
12. J. Camenisch, M. Stadler: Efficient Group Signature Schemes for Large Groups. CRYPTO '97, LNCS 1296, pp. 410–424.
13. D. Chaum, E. van Heyst: Group Signatures. EUROCRYPT '91, LNCS 547, pp. 257–265.
14. Cynthia Dwork, Moni Naor, Amit Sahai: Concurrent Zero-Knowledge. STOC 1998: 409-418.
15. Jun Furukawa, Hideki Imai: Efficient Group Signature Scheme from Bilinear Maps (with appendixes). Available from the first author via e-mail.
16. Jun Furukawa, Shoko Yonezawa: Group Signatures with Separate and Distributed Authorities. Fourth Conference on Security in Communication Networks '04 (SCN04), 2004.
17. Tetsuya Izu, Tsuyoshi Takagi: Efficient Computations of the Tate Pairingfor the Large MOV Degrees. ICISC 2002: 283-297.
18. Aggelos Kiayias, Moti Yung: Group Signatures: Provable Security, Efficient Constructions and Anonymity from Trapdoor-Holders. Cryptology ePrint Archieve, Report 2004/076.
19. A. Kiayias, Y. Tsiounis, M. Yung: Traceable Signatures. EUROCRYPT 2004, LNCS 3027, pp. 571–589.
20. Joe Kilian, Erez Petrank: Identity Escrow. CRYPTO 1998: 169-185.
21. Seungjoo Kim, Sung Jun Park, Dongho Won: Convertible Group Signatures. ASIACRYPT 1996: 311-321.
22. Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, Stefan Wolf: Pseudonym Systems. Selected Areas in Cryptography 1999: 184-199.
23. A. Menezes, C. van Oorschot, S. Vanstone: *Handbook of Applied Cryptography*, CRC Press, pp. 617-627, (1997).
24. Daniele Micciancio, Erez Petrank: Efficient and Concurrent Zero-Knowledge from any public coin HVZK protocol. Electronic Colloquium on Computational Complexity (ECCC)(045):(2002).
25. S. Mitsunari, R. Sakai, M. Kasahara: A new Traitor tracing. IEICE Trans. Fundamentals, E85-A(2), pp.481-484, Feb. 2002.
26. Atsuko Miyaji, Masaki Nakabayashi, Shunzou Takano: New explicit conditions of elliptic curve traces for FR-reduction. IEICE Trans. E85-A(2), pp. 481-484, 2002.
27. Lan Nguyen, Rei Safavi-Naini: Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings. Asiacrypt 2004. pp. 372-386.
28. Moni Naor, Moti Yung: Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. STOC 1990: 427-437.
29. D. Pointcheval, J. Stern: Security Arguments for Digital Signatures and Blind Signatures. J. Cryptology 13(3): 361-396 (2000).

30. Michael Scott, Paulo S.L.M Barreto: Generating more MNT elliptic curves. Cryptology ePrint Archieve, Report 2004/058.
31. Claus-Peter Schnorr, Markus Jakobsson: Security of Signed ElGamal Encryption. ASIACRYPT 2000: 73-89.