A Fully-Functional group signature scheme over only known-order group

Atsuko Miyaji and Kozue Umeda

1-1, Asahidai, Tatsunokuchi, Nomi, Ishikawa, 923-1292, Japan
{kozueu, miyaji}@jaist.ac.jp

Abstract. The concept of group signature allows a group member to sign message anonymously on behalf of the group. In the event of a dispute, a designated entity can reveal the identity of a signer. Previous group signature schemes use an RSA signature based membership certificate and a signature based on a proof of knowledge(SPK) in order to prove the possession of a valid membership certificate. In these schemes, SPK is generated over an unknown-order group, which requires more works and memory compared with a publicly-known-order group. Recently, a group signature based on a known-order group is proposed. However, it requires an unknown-order group as well as a known-order group. Furthermore, unfortunately, it does not provide the function of revocation. In this paper, we propose the group signature scheme based on only publicly-known-order groups. Our scheme improves the Nyberg-Rueppel signature to fit for generating membership certificates and uses SPKs over a cyclic group whose order is publicly known. As a result, our scheme reduces the size of group signature and the computational amount of signature generation and verification.

1 Introduction

A group signature proposed by Chaum and van Heyst[10], allows a group member to sign messages anonymously on behalf of the group. A group signature has a feature of tracing, that is, the identity of a signer can be revealed by a designated entity in case of dispute. A group signature consists of three entities: group members, a group manager, and an escrow manager. The group manager is responsible for the system setup, registration and revocation of group members. The escrow manager has an ability of revealing the anonymity of signatures with the help of a group manager.

A group signature consists of six functions, setup, registration of a user, revocation of a group member, signature generation, verification, and tracing, which satisfy the following features:

Unforgeability : Only group members are able to generate a signature on a message;

Exculpability : Even if the group manager, the escrow manager, and some of group members collude, they can not generate a signature on behalf of other group members;

Anonymity : Nobody can identify a group member who generated a signature on a message;

Traceability : In the case of a dispute, the identity of a group member is revealed by the cooperation of both the group manager and the escrow manager;

Unlinkability : Nobody can decide whether or not two signatures have been issued by the same group member;

Revocability : In the case of withdrawal, the group manager can revoke a member, and a signature generated by the revoked member can not pass the verification;

Anonymity after revocation : Nobody can identify a group member who generated a signature on a message even after a group member was revoked;

Unlinkability after revocation : Nobody can decide whether or not two signatures have been issued by the same group member even after a group member was revoked.

The efficiency of a group signature scheme is considered by the size of public key and signature, the work complexity of signature generation and verification, and administration complexity of revocation and registration of a group member.

Various group signature schemes have been proposed[5, 6, 9, 8, 1, 4, 16, 3, 7, 2]. These group signature schemes are classified into two types, a *public-key-registration* type, and a *certificate-based* type. In the former type, [5, 6] are constructed by using only known-order groups. However, in their schemes, both a group public key and the signature size depend on the number of group members. It yields a serious problem for large groups. In the latter type, [9, 8, 1, 4, 16, 7, 3, 2] give a membership certificate to group members, and the group signature is based on the zero-knowledge proof of knowledge(SPK) of membership certificate. Therefore, neither a group public key nor signature size depends on the number of group members. In these previous certificate-based type group signature schemes, the membership certificate has used an RSA signature over an unknown-order group, and, thus, the size of group signature becomes huge.

In this paper, we present an efficient group signature scheme based on a Nyberg-Rueppel signature. This is the first scheme that is constructed on only known-order groups and that realizes the full features of unforgeability, exculpability, anonymity, traceability, unlinkability, and revocability. As a result, the signature size and computation amount of signature generation and verification are reduced. We also give the security proof of membership certificate and group signature. Furthermore, our scheme also applies the *Certificate Revocation List*(*CRL*)*-based* revocation which proposed by Ateniese and Tsudik[3] with a slightly few additional work.

This paper is organized as follows. In the next section, we provide an overview of related work. In Section 2, we summarize some notations and definitions used in this paper. In Section 3, we propose our new group signature scheme. Section 4 discusses the security of our scheme. Features and efficiency of our scheme are analyzed in Section 5. Finally, Section 6 concludes our paper.

1.1 Related work

Various certificate-based type group signature schemes have been proposed in [1, 3, 4, 7–9, 16]. These schemes are based on the following mechanisms. A user, denoted by M_i, who wants to join the group, chooses a random secret key x_i, and computes $y_i = f(x_i)$, where f is a suitable one-way function. M_i commits to y_i (for instance, M_i signed on y_i) and sends both y_i and the commitment to the group manager denoted by GM, who returns M_i with a membership certificate $cer_i = \mathrm{Sig}_{\mathrm{GM}}(y_i)$. To sign a message m on behalf of the group, M_i encrypts y_i to c_i using the public key of the escrow manager denoted by EM, and generates a signature based on the proof of knowledge which shows the knowledge of both x_i and cer_i such that $cer_i = \mathrm{Sig}_{\mathrm{GM}}(f(x_i))$. The verification is done by checking the signature of knowledge. The escrow manager can easily reveal the anonymity of a group signature by decrypting c_i.

These group signature schemes are classified into two types, a *public-key-registration* type and a *certificate-based* type. Public-key-registration type group signature schemes[5, 6] use only known-order groups and can easily realize the revocation by removing the group member's public key. However, both a group public key and the signature size depend on the number of group members. It becomes serious if we apply them on large group. On the other hand, the group signature schemes of certificate-based type must make the member's certificate invalid when they revoke member. However, since the previous schemes [9, 8, 1, 2] do not provide any function of revocation, they can not realize the feature of revocability. The schemes [4, 16, 3, 7] provide the function of revocation. In Song's scheme[16], a membership certificate is valid for a limited period. Therefore, each group member has to update his/her membership certificate in each time period. Camenisch and Lysyanskaya's scheme[7] needs to update a membership certificate in both cases of registration and revocation. Thus, their scheme requires additional cost to manage the valid member although their verification does not depend on the number of registered or revoked member. Bresson and Stern's scheme[4] uses a CRL to realize revocation. CRL is a public list of information related with revoked-member certificates. This scheme does not have to update a membership certificate, but the size of group signature and the cost of signature generation and verification depends on the number of revoked members. Ateniese and Tsudik proposed quasi-efficient solution for CRL-based revocation[3]. CRL-based revocation scheme is based on the following mechanisms. The group manager computes $V_j = f'(cer_j)$ for each revoked member M_j by using a suitable one-way function f' and publishes V_j together with the current CRL. In the signing phase, a signer M_i also sends $T = f''(f'(cer_i))$ with a signature by using a suitable one-way function f''. In the verification phase, a verifier checks that $T \neq f''(V_j)$ for $^{\forall}V_j \in \mathcal{CRL}$. The signature size and the cost of signature generation does not depend on the number of revoked members, but the cost of verification depends on the number of revoked members. To sum up, there are certificate-update-based revocation and CRL-based revocation. In the former, the cost of verification does not depend on the number of revoked members, but each group member needs to update a membership certificate. In the latter, each

group member does not need to update a membership certificate, but the cost of verification depends on the number of revoked members.

In the certificate-based type group signature schemes, the membership certificate has used an RSA signature over an unknown-order group, and thus the size of group signature becomes huge. Recently, Nyberg-Rueppel signature was applied to a group signature[2]. However, their scheme requires an unknown-order group and must hide the membership certificate by a random value in order to satisfying the feature of anonymity and unlinkability. Thus, although a known-order group is introduced, it suffers from much work complexity and complicated interaction. Furthermore, since it does not provide the function of revocation, much administrative complexity might be required in order to revoke a member.

1.2 Our contribution

Our proposed scheme is constructed on only known-order groups and that realizes full feature of unforgeability, exculpability, traceability, unlinkability, and revocability. In our scheme, a membership certificate is generated by Nyberg-Rueppel signature, and the features of anonymity and unlinkability are realized by zero-knowledge proof of knowledge which does not have to be hidden by a random value in contrast to [2]. Thus, our group signature is rather simple than [2]. As a result, the signature size and computation amount of signature generation and verification are reduced from [2]. Furthermore, our scheme also provides the CRL-based revocation with a slightly few additional work to group members. We also give the security proof of membership certificate and group signature.

2 Preliminaries

2.1 Notation

In this section, we summarize facts used in this paper. Let the empty string be $\tilde{0}$. For a set A, $a \in_R A$ means that a is chosen randomly and uniformly from A, and $A \setminus \{a\}$ means that $A - \{a\} = \{x \in A | x \neq a\}$. For a group $G \ni g$, ord(g) means order of g in G. The bit length of a is denoted by $|a|$. Let $c[j]$ be the j-th bit of a string c. We use a collision resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^k$.

2.2 Proof of knowledge

A signature based on a zero-knowledge proof of knowledge(SPK), denoted by $SPK\{(\alpha_1, \cdots, \alpha_w) : Predicates\}$, is used for proving that a signer knows $\alpha_1, \cdots, \alpha_w$ satisfying $Predicates$. We borrow three SPKs over known-order groups from [11, 15, 6], SPK of representations and a double discrete logarithm.

Let q, p and $\tilde{p}$ be primes with $q|(p-1)$ and $p|(\tilde{p}-1)$. We use two cyclic groups $\mathbb{G}_p$ of order q with $\mathbb{G}_p \subset \mathbb{Z}_p^*$ and $\mathbb{G}_{\tilde{p}}$ of order p with $\mathbb{G}_{\tilde{p}} \subset \mathbb{Z}_{\tilde{p}}^*$.

Definition 1. *Let $g_1, \cdots, g_u$, $y_1, \cdots, y_v \in \mathbb{G}_p$. An SPK proving the knowledge of representations of $y_1, \cdots, y_v$ to the base $g_1, \cdots, g_u$ on a message $m \in \{0,1\}^*$ is denoted as*

$$SPK\{(\alpha_1, \cdots, \alpha_w) : \ y_1 = \prod_{j=1}^{J_1} g_{b_{1j}}^{\alpha_{a_{1j}}} \bmod p \ \wedge \ \cdots \ \wedge \ y_v = \prod_{j=1}^{J_v} g_{b_{vj}}^{\alpha_{a_{vj}}} \bmod p \ \}(m),$$

where $J_i \in [1, \cdots, u]$ are the number of bases of y_i, $a_{ij} \in [1, \cdots, w]$ are indexes of the elements $\alpha_{a_{ij}}$, and $b_{ij} \in [1, \cdots, u]$ are indexes of the bases $g_{b_{ij}}$, which consists of a set of $(c, s_1, \cdots, s_w) \in \{0,1\}^k \times \mathbb{Z}_q^w$ satisfying $c = \mathcal{H}(g_1 || \cdots || g_u || y_1 || \cdots || y_v || y^c \prod_{j=1}^{J_1} g_{b_{1j}}^{s_{a_{1j}}} \bmod p || \cdots || y_v^c \prod_{j=1}^{J_v} g_{b_{vj}}^{s_{a_{vj}}} \bmod p || m)$.

If a signer knows $x_1, \cdots, x_w \in \mathbb{Z}_q$ such that $y = \prod_{j=1}^{J_1} g_{b_{1j}}^{x_{a_{1j}}} \bmod p$, $\cdots$, $y_v = \prod_{j=1}^{J_v} g_{b_{vj}}^{x_{a_{vj}}} \bmod p$, then a signature on a message m can be computed as follows:

1. choose random exponents $r_d \in \mathbb{Z}_q^*$ for $1 \leq d \leq w$,
2. compute $c = \mathcal{H}(g_1 || \cdots || g_u || y_1 || \cdots || y_v || \prod_{j=1}^{J_1} g_{b_{1j}}^{r_{a_{1j}}} \bmod p || \cdots || \prod_{j=1}^{J_v} g_{b_{vj}}^{r_{a_{vj}}} \bmod p || m)$ and
3. compute $s_d = r_d - c x_d \bmod q$ for $1 \leq d \leq w$.

Definition 2. *Let $\tilde{g}$, $\tilde{y} \in \mathbb{G}_{\tilde{p}}$ and $g \in \mathbb{G}_p$. An SPK proving the knowledge of double discrete logarithm of $\tilde{y}$ to the base $\tilde{g}$ and g on a message $m \in \{0,1\}^*$ is denoted as*

$$SPK\{(\alpha) : \tilde{y} = \tilde{g}^{g^\alpha} \bmod \tilde{p} \ \}(m),$$

which consists of a set of $(c, s_1, \cdots, s_k) \in \{0,1\}^k \times \mathbb{Z}_q^k$ satisfying $c = \mathcal{H}(g || \tilde{g} || \tilde{y} || (\tilde{y}^{c[1]} \tilde{g}^{1-c[1]})^{g^{s_1}} \bmod \tilde{p} || \cdots || \ (\tilde{y}^{c[k]} \tilde{g}^{1-c[k]})^{g^{s_k}} \bmod \tilde{p} || m)$.

A signer who knows the secret key $x \in \mathbb{Z}_q$ with $\tilde{y} = \tilde{g}^{g^x} \bmod \tilde{p}$ can compute a signature $(c, s_1, \cdots, s_k) = SPK\{(\alpha) : \tilde{y} = \tilde{g}^{g^\alpha} \bmod \tilde{p} \ \}(m)$ on a message m as follows:

1. choose random exponents $r_j \in \mathbb{Z}_q^*$ for $1 \leq j \leq k$,
2. compute $c = \mathcal{H}(g || \tilde{g} || \tilde{y} || \tilde{g}^{g^{r_1}} \bmod \tilde{p} || \cdots || \tilde{g}^{g^{r_k}} \bmod \tilde{p} || m)$, and
3. compute $s_j = r_j - c[j] x \bmod q$ for $1 \leq j \leq k$.

3 Proposed scheme

We present the group signature scheme based on a Nyberg-Rueppel signature after we define a new SPK and a new problem based on DLP, and modify the Nyberg-Rueppel signature.

3.1 New SPK of a common discrete logarithm over different groups

Let us define a new SPK which proves the knowledge of a common discrete logarithm over different groups. Let P be a product pq of prime p and $q|(p-1)$, $\tilde{P}$ be a prime with $P|(\tilde{P}-1)$. We also use two cyclic groups $\mathbb{G}_P$ of order q with $\mathbb{G}_P \subset \mathbb{Z}_P^*$ and $\mathbb{G}_{\tilde{P}}$ of order P with $\mathbb{G}_{\tilde{P}} \subset \mathbb{Z}_{\tilde{P}}^*$.

Definition 3 (SPK of a common discrete logarithm over different groups). *Let $g, y \in \mathbb{G}_P$ with ord(g) = ord(y) and $\tilde{g}, \tilde{y} \in \mathbb{G}_{\tilde{P}}$ with ord($\tilde{g}$) = ord($\tilde{y}$). An SPK proving the knowledge of common discrete logarithm of y to the base g and $\tilde{y}$ to the base $\tilde{g}$ on a message $m \in \{0,1\}^*$ is denoted as*

$$SPK\{(\alpha) : y = g^\alpha \bmod P \ \wedge \ \tilde{y} = \tilde{g}^\alpha \bmod \tilde{P} \ \wedge \ \alpha \in \mathbb{Z}_P \ \}(m),$$

which consists of a set of $(c,s) \in \{0,1\}^k \times \mathbb{Z}_P$ satisfying $c = \mathcal{H}(g||y||\tilde{g}||\tilde{y}||$ $y^c g^s \bmod P||\tilde{y}^c\tilde{g}^s \bmod \tilde{P}||m)$.

If a signer knows such an integer $x \in \mathbb{Z}_P$ that both $y = g^x \bmod P$ and $\tilde{y} = \tilde{g}^x \bmod \tilde{P}$ hold, a signature on a message m corresponding to public keys y and $\tilde{y}$ can be computed as follows:

1. choose a random exponent $r \in \mathbb{Z}_P^*$,
2. compute $c = \mathcal{H}(g||y||\tilde{g}||\tilde{y}||g^r \bmod P||\tilde{g}^r \bmod \tilde{P}||m)$, and
3. compute $s = r - cx \bmod P$.

Lemma 1. *The interactive protocol corresponding to $SPK\{(\alpha) : y = g^\alpha \bmod P \ \wedge \ \tilde{y} = \tilde{g}^\alpha \bmod \tilde{P} \ \wedge \ \alpha \in \mathbb{Z}_P \ \}(m)$ is a honest-verifier perfect zero-knowledge proof of knowledge of common discrete logarithm of y to the base g and $\tilde{y}$ to the base $\tilde{g}$.*

Proof : The proof on the perfect zero-knowledge part is quite standard. We restrict our attention to the proof of knowledge part. By using the fact that the equivalent protocol[15] is a proof of knowledge, it is sufficient to show that the knowledge extractor can compute the witness once he has found two accepting sets (t_1, t_2, c, s) and (t_1, t_2, c', s'). Since both $t_1 = y^c g^s = y^{c'} g^{s'} \pmod P$ and $t_2 = \tilde{y}^c\tilde{g}^s = \tilde{y}^{c'}\tilde{g}^{s'} \pmod{\tilde{P}}$ hold, we have $y = g^{\frac{s'-s}{c-c'}} \pmod P$ and $\tilde{y} = \tilde{g}^{\frac{s'-s}{c-c'}} \pmod{\tilde{P}}$. From these equations, we have

$$\begin{cases} x_q = \frac{s'-s}{c-c'} \bmod q, \\ x_p = \frac{s'-s}{c-c'} \bmod p \end{cases} .$$

On the other hand, we can compute such an integer $x \in \mathbb{Z}_P$ that

$$\begin{cases} x \equiv x_q \bmod q \\ x \equiv x_p \bmod p \end{cases}$$

by using Chinese Remainder Theorem. Then both $y = g^x \bmod P$ and $\tilde{y} = \tilde{g}^x \bmod \tilde{P}$ hold. Therefore, $SPK\{(\alpha) : y = g^\alpha \bmod P \ \wedge \ \tilde{y} = \tilde{g}^\alpha \bmod \tilde{P} \ \wedge \ \alpha \in \mathbb{Z}_P \ \}(m)$ is a honest-verifier perfect zero-knowledge proof of knowledge of common discrete logarithm of y to the base g and $\tilde{y}$ to the base $\tilde{g}$. □

3.2 The multiple discrete logarithm problem

Before presenting our scheme, we define the Multiple Discrete Logarithm Problem(MDLP), which is used for the security proof of our scheme. Let k be a security parameter, q and p be primes with $|q| = k$ and $q|(p-1)$, P be a product of q and p, g_1, g_2 and g_3 be elements in $\mathbb{Z}_P^*$ with order q.

Problem 1 (MDLP) *Given $\mathbb{Z}_P$ and g_1, g_2 and $g_3 \in \mathbb{Z}_P^*$ with order q such that the discrete logarithms based on each other element are unknown, find a pair $(x_1, x_2, x_3) \in \mathbb{Z}_P \times \mathbb{Z}_q \times \mathbb{Z}_q$ such that $x_1 g_1^{x_1} g_2^{x_2} = g_3^{x_3} \pmod P$.*

Assumption 1 (MDL Assumption) *There is no probabilistic polynomial-time algorithm* P *that can solve the Problem 1.*

3.3 The modified Nyberg-Rueppel signature scheme

Let us summarize the original Nyberg-Rueppel signature scheme[14]. For a q-order element $g \in \mathbb{Z}_p^*$, a signer chooses his secret key $x \in_R \mathbb{Z}_q$ and computes his public key $y = g^x \bmod p$. A signature $(r, s) \in \mathbb{Z}_p \times \mathbb{Z}_q$ on a message $m \in \mathbb{Z}_p^*$ is computed as $r = mg^{-w} \bmod p$ and $s = w - rx \bmod q$ for a random integer $w \in_R \mathbb{Z}_q$, which is verified by recovering the message m as $m = ry^r g^s \bmod p$.

Message recovery signature schemes are subject to an existential forgery, in which an attacker cannot control a message. In a sense, it is not a serious problem because we can avoid such a forgery by restricting a message to a particular format. However, suppose that we want to use it for a membership certificate of DLP-based key like $m = g^t \bmod p$. Then, by using a valid signature for a message $m = g^t \bmod p$ with a known discrete logarithm t, it is easy to obtain a forged signature for some known message $m' = g^{t'} \bmod p$, in which an attacker can control a message of m'. Therefore, we must remove such a defect from the original Nyberg-Rueppel signature to generate a membership certification of a DLP-based key.

In order to generate a membership certificate of a DLP-based key securely, we introduce another base $h \in \mathbb{Z}_p^*$ with order q such that the discrete logarithm of h to the base g is unknown. We restrict the message space for Nyberg-Rueppel signature to $\{h^t \bmod p \mid t \in \mathbb{Z}_q\}$. In our scheme, GM or M_i computes each public key as $y = g^{x_{\mathrm{GM}}} \bmod p$ or $z_i = h^{x_i} \bmod p$, respectively. Then, a membership certificate $(r_i, s_i) \in \mathbb{Z}_p \times \mathbb{Z}_q$ of M_i's public key $z_i = h^{x_i} \bmod p$ is given as $z_i = r_i y^{r_i} g^{s_i} \pmod p$.

3.4 Functional description

A group signature scheme with CRL-based revocation consists of the following procedures:

Setup: A probabilistic polynomial-time algorithm that on input a security parameter k outputs the group public key $\mathcal{Y}$ (including all system parameters), the secret key $\mathcal{S}$ of the group manager, and the initial certificate revocation list $\mathcal{CRL}$.

Registration: A protocol between the group manager and a user that registers a user as a new group member. The group manager outputs the renewed member list $\mathcal{ML}$. The user outputs a membership key with a membership certificate.

Revocation: A probabilistic polynomial-time algorithm that on input the renewed revoked member list $\mathcal{RML}$ outputs a renewed certificate revocation list $\mathcal{CRL}$ corresponding to $\mathcal{RML}$.

Sign: A probabilistic polynomial-time algorithm that on input a group public key $\mathcal{Y}$, a membership key, a membership certificate, and a message m outputs a group signature σ.

Verification: A boolean-valued algorithm that on input a message m, a group signature σ, a group public key $\mathcal{Y}$, and a current certificate revocation list $\mathcal{CRL}$ returns 1 if and only if σ was generated by some valid group member.

Tracing: An algorithm that on input a valid group signature σ, a group public key $\mathcal{Y}$, the group manager's secret key, and the member list $\mathcal{ML}$ outputs the identity of a signer.

3.5 Scheme intuition

Our scheme must permit M_i to prove knowledge of his membership certificate (r_i, s_i) corresponding his membership key x_i without revealing any information of x_i, r_i or s_i. However, there has not been any SPK which proves the knowledge of the membership certificate directly. So, we modify Nyberg-Rueppel signature as follows. Let $\tilde{P}$ be a prime with $P|(\tilde{P}-1)$, $P = pq$ and $q|(p-1)$ and q-order elements g_1 and $g_2 \in \mathbb{Z}_P^*$. GM issues a membership certificate (A_i, b_i) of M_i's public key $z_i = g_2^{x_i} \bmod P$ as $g_2^{x_i} = A_i y^{A_i} g_1^{b_i} \pmod P$. This exactly means that our membership certificate is based on MDLP. To forge a valid membership certificate is equivalent to solve MDLP. Under the Assumption 1, it is difficult to find a set of $\{x_i, (A_i, b_i)\}$ such that $g_2^{x_i} = A_i y^{A_i} g_1^{b_i} \bmod P$ without knowing the discrete logarithm of g_1, g_2 and y based on each other elements. Therefore, the membership certificate (A_i, b_i) corresponding to a membership key x_i can be obtained by only the interactive protocol between GM and M_i. In the signing phase, we employ a base $\tilde{g} \in \mathbb{Z}_{\tilde{P}}^*$ with order P to protect any information of the membership certificate (A_i, b_i) and corresponding membership key x_i, M_i computes a random base $T = \tilde{g}^W \bmod \tilde{P}$ for a random integer $W \in_R \mathbb{Z}_P$ and generates a signature based on the proof of knowledge of $\{x_i, (A_i, b_i)\}$ such that $T^{g_2^{x_i}} = T^{A_i y^{A_i} g_1^{b_i}} \bmod \tilde{P}$ holds. This can be constructed by using SPK which defined in Section 2.2.

3.6 Our group signature scheme

We present a new group signature scheme with CRL-based revocation, which uses only known-order groups. Let k be the security parameter and the initial member list $\mathcal{ML}$, the initial revoked member list $\mathcal{RML}$ and the initial membership certificate revocation list $\mathcal{CRL}$ be null.

Setup(k)

1. Choose a random k-bit prime q, a random prime p of such that $q|(p-1)$ and set $P = pq$.
2. Choose a random prime $\tilde{P}$ of such that $P|(\tilde{P}-1)$.
3. Set each cyclic subgroup $\mathbb{G}_P \subset \mathbb{Z}_P^*$ with order q and $\mathbb{G}_{\tilde{P}} \subset \mathbb{Z}_{\tilde{P}}^*$ with order P.
4. Choose random elements g_1, g_2, g_3 and $g_4 \in_R \mathbb{G}_P \setminus \{1\}$ such that the discrete logarithms based on each other elements are unknown.
5. Choose a random element $\tilde{g} \in_R \mathbb{G}_{\tilde{P}} \setminus \{1\}$.
6. Compute $y_1 = g_1^{x_{\mathrm{GM}}} \bmod P$ and $y_2 = g_3^{x_{\mathrm{GM}}} \bmod P$ for a secret key $x_{\mathrm{GM}} \in_R \mathbb{Z}_q$.
7. Output the group public key $\mathcal{Y} = \{q, P, \tilde{P}, g_1, g_2, g_3, g_4, \tilde{g}, y_1, y_2\}$ and the secret key $\mathcal{S} = \{x_{\mathrm{GM}}\}$.

Registration($\mathcal{Y}$, $\mathcal{S}$, $\mathcal{ML}$)

1. M_i chooses a membership key $x_i \in_R \mathbb{Z}_q$, sets $z_i = g_2^{x_i} \bmod P$, and sends z_i with $\sigma_i = SPK\{(\alpha) : z_i = g_2^{\alpha} \bmod P\ \}(\tilde{0})$ to GM[1].
2. GM checks the validity of σ_i, chooses a random integer $w_i \in_R \mathbb{Z}_q$, computes $A_i = z_i g_1^{-w_i} \bmod P$ and $b_i = w_i - A_i x_{\mathrm{GM}} \bmod q$, and sends $(A_i, b_i) \in \mathbb{Z}_P \times \mathbb{Z}_q$ to M_i through a secure cannel.
3. GM adds (A_i, b_i) with M_i's identity ID_i to the member list $\mathcal{ML}$.
4. M_i verifies that $A_i y_1^{A_i} g_1^{b_i} = z_i \pmod P$.
5. GM outputs the renewed member list $\mathcal{ML} = \{(ID_i, A_i, b_i)\}$.
6. M_i possesses a membership key x_i and a membership certificate $(A_i, b_i) \in \mathbb{Z}_P \times \mathbb{Z}_q$.

In order to revoke a new subset of members whose revoked member list is $\mathcal{RML} = \{(ID, b)\}$ with $|\mathcal{RML}| = u$, GM renews the certificate revocation list $\mathcal{CRL}$ by running the following Revocation protocol.

Revocation($\mathcal{RML}$)

1. Choose a new revocation base $g_4 \in_R \mathbb{G}_P \setminus \{1\}$ and update $\mathcal{Y}$.
2. Compute $V_j = g_4^{b_j} \bmod P$ for $b_j \in \mathcal{RML}$ $(1 \le j \le u)$.
3. Output the renewed certificate revocation list $\mathcal{CRL} = \{V_j \mid 1 \le j \le u\}$.

Sign($\mathcal{Y}$, g_4 x_i, A_i, b_i, m)

1. Choose a random integer $w \in_R \mathbb{Z}_q$.
2. Compute $T_1 = \tilde{g}^{g_3^w} \bmod \tilde{P}$, $T_2 = T_1^{g_4^{b_i}} \bmod \tilde{P}$, $T_3 = g_3^{b_i} g_4^w \bmod P$, $T_4 = A_i g_3^w \bmod P$, and $T_5 = y_2^w \bmod P$.
3. Generate

$$\sigma_1 = SPK\{(\alpha_1, \alpha_2) :\ T_1 = \tilde{g}^{g_3^{\alpha_2}} \bmod \tilde{P} \ \wedge\ T_2 = T_1^{g_4^{\alpha_1}} \bmod \tilde{P} \ \wedge\ T_3 = g_3^{\alpha_1} g_4^{\alpha_2} \bmod P\ \}(m)$$

$$= (c_1, s_{11}, \cdots, s_{1k}, s_{21}, \cdots, s_{2k}) \in \{0,1\}^k \times \mathbb{Z}_q^{2k}$$

as follows:

[1] We can also add an interactive protocol to make a member's secret key jointly by a member and GM.

- choose random integers ω_{1j}, $\omega_{2j} \in_R \mathbb{Z}_q$ for $1 \leq j \leq k$,
- compute
  - $t_{1j} = \tilde{g}^{g_3^{\omega_{2j}}} \bmod \tilde{P}$, $t_{2j} = T_1^{g_4^{\omega_{1j}}} \bmod \tilde{P}$, and $t_{3j} = g_3^{\omega_{1j}} g_4^{\omega_{2j}} \bmod P$ for $1 \leq j \leq k$,
  - $c_1 = \mathcal{H}(g_3||g_4||\tilde{g}||T_1||T_2||T_3||t_{11}||\cdots||t_{1k}||t_{21}||\cdots||t_{2k}||t_{31}||\cdots||t_{3k}||m)$,
  - $s_{1j} = \omega_{1j} - c_1[j]b_i \bmod q$ and $s_{2j} = \omega_{1j} - c_1[j]w \bmod q$ for $1 \leq j \leq k$.

4. Generate

$$\sigma_2 = SPK\{(\alpha_3, \alpha_4, \alpha_5, \alpha_6) : \ \alpha_3 \in \mathbb{Z}_P \ \wedge \ T_3 = g_3^{\alpha_4} g_4^{\alpha_6} \bmod P \wedge$$
$$T_4 = y_1^{-\alpha_3} g_1^{-\alpha_4} g_2^{\alpha_5} g_3^{\alpha_6} \bmod P \ \wedge \ T_5 = y_2^{\alpha_6} \bmod P \ \wedge \ \tilde{g}^{T_4} = T_1^{\alpha_3} \bmod \tilde{P} \ \}(m)$$
$$= (c_2, s_3, s_4, s_5, s_6) \in \{0,1\}^k \times \mathbb{Z}_q^3 \times \mathbb{Z}_P$$

as follows:
- choose $\omega_3 \in_R \mathbb{Z}_P$, $\omega_4, \omega_5, \omega_6 \in_R \mathbb{Z}_q$,
- compute
  - $t_4 = g_3^{\omega_4} g_4^{\omega_6} \bmod P$, $t_5 = y_1^{-\omega_3} g_1^{-\omega_4} g_2^{\omega_5} g_3^{\omega_6} \bmod P$, $t_6 = y_2^{\omega_6} \bmod P$, and $t_7 = T_1^{\omega_3} \bmod \tilde{P}$,
  - $c_2 = \mathcal{H}(g_1||g_2||g_3||g_4||\tilde{g}||y_1||y_2||T_1||T_3||T_4||T_5||t_4||t_5||t_6||t_7||m)$,
  - $s_3 = \omega_3 - c_2 A_i \bmod P$, $s_4 = \omega_4 - c_2 b_i \bmod q$, $s_5 = \omega_5 - c_2 x_i \bmod q$ and $s_6 = \omega_6 - c_2 w \bmod p$.

5. Output a group signature $\sigma = \{T_1, T_2, T_3, T_4, T_5, \sigma_1, \sigma_2\}$.

Verification($\mathcal{Y}$, $\mathcal{CRL}$, m, σ)
1. Check the validity of σ_1 and σ_2.
2. If $T_1^{V_j} \neq T_2 \bmod \tilde{P}$ for $^\forall V_j \in \mathcal{CRL}$, then accept the signature otherwise reject the signature.

Tracing(x_{GM}, $\mathcal{ML}$, σ)
1. Recover A_i by $A_i = T_4 / T_5^{1/x_{\mathrm{GM}}} \bmod P$.
2. Identify a signer M_i from A_i by using the member list $\mathcal{ML}$.
3. Output the signer's identity ID_i.

In our scheme, in order to realize the features of anonymity and unlinkability, GM has to keep $\mathcal{ML}$ secretly and send a membership certificate to a group member through a secure cannel. This assumption is required in the CRL-based revocation as in [3]. To reduce the features of anonymity and unlinkability to GM, GM may be separated to two managers, the group manager and the escrow manager by applying techniques of multi-party computation to generate a membership certificate.

4 Security consideration

We use two different signature schemes in our group signature scheme. One is the modified Nyberg-Rueppel signature scheme that generates the membership certificate, and the other is SPK that generates the group signature. In this section, we consider the security of a membership certificate and the group signature.

4.1 Security proof on the membership certificate

The security of the membership certificate in our scheme is based on the difficulty of the MDLP. We show the membership certificate is secure against any probabilistic polynomial-time adversaries.

Let us define one more security assumption. For the security parameter k, primes p and q with $|q| = k$ and $q|(p-1)$, $P = pq$ and g_1, g_2, $g_3 \in \mathbb{Z}_P^*$ with order q, a set of solutions of Problem 1 is denoted as

$$\mathcal{X}(\mathbb{Z}_P, g_1, g_2, g_3) = \{(x_1, x_2, x_3) \in \mathbb{Z}_P \times \mathbb{Z}_q \times \mathbb{Z}_q \mid x_1 g_1^{x_1} g_2^{x_2} = g_3^{x_3} \pmod P\}$$

where the discrete logarithms of g_1, g_2, and g_3 based on each other element is not known.

Problem 2 (Strong-MDLP) *Given $\mathbb{Z}_P$, g_1, g_2, and $g_3 \in \mathbb{Z}_P^*$ such that the discrete logarithm based on each other element is not known and any subset $X \subset \mathcal{X}(\mathbb{Z}_P, g_1, g_2, g_3)$ with the polynomial order $|X|$, find a pair $(x_1, x_2, x_3) \in \mathbb{Z}_P \times \mathbb{Z}_q \times \mathbb{Z}_q$ such that $x_1 g_1^{x_1} g_2^{x_2} = g_3^{x_3} \pmod P$ and $(x_1, x_2, x_3) \notin X$.*

Assumption 2 (Strong-MDLP Assumption) *There is no probabilistic polynomial-time algorithm* P *that can solve the Problem 2.*

More formally, the following experiment is executed with algorithm A.

`Break-strong-MDLP`$(\mathsf{A}, k, q, P, g_1, g_2, g_3)$

1. `Choose a polynomial-order subset` $X \subset \mathcal{X}(\mathbb{Z}_P, g_1, g_2, g_3)$.
2. $(x_1, x_2, x_3) \leftarrow \mathsf{A}^X(k, g_1, g_2, g_3, q, P)$.
3. `If` $(x_1, x_2, x_3) \in \mathbb{Z}_P \times \mathbb{Z}_q \times \mathbb{Z}_q$, $g_3^{x_3} = x_1 {g_1}^{x_1} {g_2}^{x_2} \pmod P$, `and` $(x_1, x_2, x_3) \notin X$
   `then return 1,`
   `else return 0.`

The strong MDLP assumption is that the maximum success probability of Break-strong-MDLP$(\mathsf{A}, k, q, P, g_1, g_2, g_3)$ over all the probabilistic polynomial-time adversary is negligible in k.

By using Assumption 2, we can formalize the security of the membership certificate as follows. Let us define A be a probabilistic polynomial-time oracle Turing machine, which gets input $\mathcal{Y}$ and runs with a *membership certificate oracle* $\mathsf{O_C}(t, \mathcal{Y}, \mathcal{S}, \cdot)$, which on input $z \in \mathbb{Z}_P^*$ outputs a membership certificate (A, b). The adversary A may query the oracle adaptively. Eventually, adversary outputs a new membership certificate (A', b') for a public key z' and the corresponding membership key x'. The adversary wins if z' was not queried and $A' {y}^{A'} {g_1}^{b'} = z' \pmod P$. More formally, the following experiment is executed with the algorithm A.

`Adversary` (A, k)

1. `Set` $(\mathcal{S}, \mathcal{Y}) \leftarrow$ `Setup`(k)
2. `Set` $(A', b', z', x') \leftarrow \mathsf{A}^{\mathsf{O_C}}(k, \mathcal{Y})$

3. If $A'y^{A'}{g_1}^{b'} \neq z' \pmod P$ or z' was queried to $\mathsf{O_C}$,
then return "adversary failed",
else return "adversary succeeded".

From the above discussion, the security of our certificate is proved as follows.

Theorem 1. *Let* A *be a probabilistic polynomial-time adversary of time complexity* τ *with at most* Q *queries to an oracle* $\mathsf{O_C}$*. If the adversary successfully forges a new certificate, then there exists an adversary* B *performing an attack against the strong MDLP with at least the same advantage. Furthermore the time complexity of* B *is at most* τ*.*

4.2 Security proof on the group signature

We show the security of the group signature.

Theorem 2. *The interactive protocol underlying the group signature scheme is a honest-verifier perfect zero-knowledge proof of knowledge of a membership certificate and corresponding membership key. Furthermore, it proves that the a pair* (T_4, T_5) *encrypts the membership certificate under the group manager's public key* y_2*.*

Proof : The proof that the perfect zero-knowledge part is quite standard. We restrict our attention to the proof of knowledge part. By the properties of the SPK protocol, the signer can produce values of $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, α_5 and α_6 such that

$$T_1 = \tilde{g}^{g_3^{\alpha_2}} \bmod \tilde{P} \tag{1}$$

$$T_2 = T_1^{g_4^{\alpha_1}} \bmod \tilde{P} \tag{2}$$

$$T_3 = g_3^{\alpha_1} g_4^{\alpha_2} = g_3^{\alpha_4} g_4^{\alpha_6} \bmod P \tag{3}$$

$$T_4 = y_1^{-\alpha_3} g_1^{-\alpha_4} g_2^{\alpha_5} g_3^{\alpha_6} \bmod P \tag{4}$$

$$T_5 = y_2^{\alpha_6} \bmod P \tag{5}$$

$$\tilde{g}^{T_4} = T_1^{\alpha_3} \bmod \tilde{P} \tag{6}$$

$$\alpha_3 \in \mathbb{Z}_P \tag{7}$$

hold, in which $\alpha_1 = \alpha_4$ and $\alpha_2 = \alpha_6$ hold from Equation (3). Thus, Equations (1) and (2) represent

$$T_1 = \tilde{g}^{g_3^{\alpha_6}} \bmod \tilde{P} \tag{8}$$

and

$$T_2 = T_1^{g_4^{\alpha_4}} \bmod \tilde{P}. \tag{9}$$

From Equations (4) and (8), we can rewrite Equation (6) as

$$\begin{aligned} & \tilde{g}^{y_1^{-\alpha_3} g_1^{-\alpha_4} g_2^{\alpha_5} g_3^{\alpha_6}} = (\tilde{g}^{g_3^{\alpha_6}})^{\alpha_3} \pmod{\tilde{P}} \\ \Leftrightarrow \quad & y_1^{-\alpha_3} g_1^{-\alpha_4} g_2^{\alpha_5} g_3^{\alpha_6} \equiv g_3^{\alpha_6 \alpha_3} \pmod P \\ \Leftrightarrow \quad & g_2^{\alpha_5} \equiv \alpha_3 y_1^{\alpha_3} g_1^{\alpha_4} \pmod P. \end{aligned} \tag{10}$$

Thus, a set of $\{\alpha_5, (\alpha_3, \alpha_4)\}$ is coincident with the valid membership certificate and corresponding membership key. From using Equation (10), Equation (4) represents

$$T_4 = \alpha_3 g_3^{\alpha_6} \pmod{P}.$$

Thus, a pair of (T_4, T_5) is an encryption of α_3 by the group manager's public key y_2. Therefore, the group signature is a honest-verifier perfect zero-knowledge proof of knowledge of a membership certificate and corresponding membership key, and it proves that the a pair (T_4, T_5) is an encryption of the membership certificate by the group manager's public key y_2. □

5 Analysis of our scheme

5.1 Features

Here we show that our scheme satisfies all features necessary for group signatures.

Unforgeability : From the proof of Theorem 2, a set of $(T_1, T_2, T_3, T_4, T_5)$ is an unconditional binding commitment to a valid membership certificate (A_i, b_i) and corresponding membership key x_i. Under the Assumption 2, it is infeasible to find a certificate (A_i, b_i) corresponding a membership key x_i without knowledge of the group manager's secret key. Therefore, only group members who have a valid membership certificate are able to generate a signature on a message;

Exculpability : GM knows a member's membership certificate, but he can not get any information about the corresponding membership key x_i. Hence, even if GM colludes with some group members, they cannot sign on behalf of M_i.

Anonymity : Assuming that the function $\mathcal{H}$ is a random function, the SPKs of σ_1 and σ_2 do not leak any information since their interactive counterparts are based on the honest-verifier perfect zero-knowledge. To decide whether some group member with certificate (A_i, b_i) generated, it is required to decide whether $\log_{\tilde{g}} T_1 = T_4/A_i$, $\log_{T_1} T_2 = g_4^{b_i}$ or $\log_{g_4} T_3/g_3^{b_i} = \log_{g_3} T_4/A_i = \log_{y_2} T_5$. However, these are impossible under the decision Diffie-Hellman assumption[12], and hence anonymity is guaranteed.

Traceability : When the signature is valid, (T_4, T_5) is coincident with the encryption of the membership certificate A_i, which can be uniquely recovered by GM. Therefore, a member can be traced in case of dispute. On the other hand, in order to impersonate another signer with (A_i', b_i'), they must forge the membership certificate (A_i', b_i'). Under the Assumption 2, it is infeasible.

Unlinkability : In order to decide whether or not two signatures $\{T_1, T_2, T_3, T_4, T_5, \sigma_1, \sigma_2\}$ and $\{T_1', T_2', T_3', T_4', T_5', \sigma_1', \sigma_2'\}$ were generated by the same group member, we need to decide whether or not $\log_{\tilde{g}} T_1/T_1' = T_4/T_4'$, $(\log_{T_1} T_2)/(\log_{T_1'} T_2') = 1$ or $\log_{g_4} T_3/T_3' = \log_{g_3} T_4/T_4' = \log_{y_2} T_5/T_5'$ holds. However, these are impossible under the decision Diffie-Hellman assumption[12], and hence group signatures are unlinkable each other.

Revocability : Each group signature must prove the knowledge of b_i with $T_2 = T_1^{g_4^{b_i}} \bmod \tilde{P}$, where GM publishes revoked member's membership certificate as $V = g_4^b \bmod \tilde{P}$. Therefore, if a signer is a revoked member (i.e., $b_i = b$), then $T_1^V = T_2 \bmod \tilde{P}$ for some V holds. The verifier can check the equation and judge whether the signer has been revoked or not. In order to forge the group signature that passes verification, a revoked member must substitute another b' for a part of membership certificate b, but it is impossible under Assumption 2. We can say that a revoked member can not generate a valid group signature.

Anonymity after revocation : A CRL certificate, however do not leak any information of group member. Therefore nobody can identify a group member who generated a signature on a message even after a group member was revoked.

Unlinkability after revocation : In order to decide whether or not two signatures σ and σ' based on different-time CRL CRL and CRL$'$ were generated by the same member M_j whose certificate is in CRL$'$, we need to decide whether or not $\log_{g_4} log_{T_1} T_2 = \log g_4' V_j'$ holds. However, this is impossible under the decision Diffie-Hellman assumption[12], and thus group signatures are unlinkable even after a group member was revoked.

5.2 Efficiency

We compare our scheme with previous schemes [3] from the viewpoints of both computational work and signature size in Table 1. Let P or q be 1200 or 160 bits, respectively. Here M denotes the computational work of a multiplication over an 1200-bit modulus. We assume the binary method or the extended binary method to compute the exponentiation or multiple exponentiations[13], respectively.

Table 1 shows that our scheme reduces both of signature size and verification work by about 1/3 than [3], maintaining the same security level. Furthermore, our scheme is slightly more efficient than even the group signature scheme based on known-order cyclic groups proposed by G. Ateniese and B. de Medeiros[2], which does not satisfy the feature of revocability as mentioned in Section 1. Although revocability can be easily added in a simple way[3], it just increases both the signature size and computational work. Our scheme is optimized under such a condition that realizes all features, including the revocability. Therefore, our scheme is much better than a scheme combined [2] with the revocation function of [3].

Since our scheme uses the SPK of double discrete logarithms, it seems to require much computational work in contrast to group signature schemes with revocation[5, 6] which do not use SPK of double discrete logarithms. However, their group public key and signature size depend on the number of group members, and thus these schemes are less efficient than our scheme for large groups like of 1000 members.

Table 1. Comparison of the efficiency

	Work		Signature Size
	Sign	Verification	Signature
[2] with [3]	2020.3×10^3 M	$(2031.3 + 1.8u) \times 10^3$ M	101.6 KByte
[5](1)	$200n + 760$M	$200(n + 1)$ M	$380 + 20n$ KByte
Our scheme	705.1×10^3 M	$(700.4 + 1.8u) \times 10^3$ M	31.3 KByte

(1) The number of group member denoted by n.

6 Conclusion

We have proposed the group signature with CRL-based revocation. In our scheme, the membership certificate is constructed by using improved Nyberg-Rueppel signature with appendix. As a result, the signature size and computational work of signature generation and verification can be reduced because all secret data can be computed by using the knowledge of order of group.

Our scheme uses the proof of knowledge involving double discrete logarithm in the same way as previous group signatures, which requires many computational work. Furthermore our scheme uses a membership certificate based on a special assumption of Multiple DLP. Developing a membership certificate based on standard assumptions is a challenging open problem. Another interesting open question is to find the relation ship among the Multiple DLP, DLP.

References

1. G. Ateniese and J. Camenisch and M. Joye and G. Tsudik, "A practical and provably secure Coalition-Resistant group signature scheme", Advances in Cryptology-Proceedings of CRYPTO2000, LNCS 1880(2000), pp. 255-270.
2. G. Ateniese and B. de Medeiros, "Efficient group signatures without trapdoors", Cryptology ePrint Archive, available from http://citeseer.nj.nec.com/ateniese02efficient.html.
3. G. Ateniese and G. Tsudik, "Quasi-efficient revocation of group signatures", In the proceeding of FC2002, 2002.
4. E. Bresson and J. Stern, "Group signatures with efficient revocation", In proceeding of PKC2001, LNCS 1992(2001), pp. 190-206.
5. J. Camenisch, "Efficient and generalized group signature", Advances in Cryptology-Proceedings of EUROCRYPT'97, LNCS 1233(1997), pp. 465-479.
6. J. Camenisch, "Group signature schemes and payment systems based on the discrete logarithm problem", PhD thesis, vol. 2 of ETH-Series in Information Security an Cryptography, Hartung-Gorre Verlag, Konstanz, 1998, ISBN 3-89649-286-1.
7. J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials", Advances in Cryptology-Proceedings of CRYPTO2002, LNCS 2442(2002), pp. 61-76.
8. J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant", preliminary version in Advances in Cryptology - ASIACRYPT'98, Tech. Rep., RS-98-27, BRICS, 1998.

9. J. Camenisch and M. Stadler, "Efficient group signature schemes for large group", Advances in Cryptology-Proceedings of CRYPTO'97, LNCS 1296(1997), pp. 410-424.
10. D. Chaum and E. van Heyst, "Group signatures", Advances in Cryptology-Proceedings of EUROCRYPT'91, LNCS 547(1991), pp. 257-265.
11. D. Chaum, J. H. Evertse and J. van de Graaf, "An improved protocol for demonstration possession of discrete logarithms and some generalizations", Advances in Cryptology-Proceedings of EUROCRYPT'87, LNCS 304(1987), pp. 127-141.
12. W. Diffie and M. E. Hellman, "New directions in cryptography", IEEE Transaction on Information Theory IT-22, 1976, pp. 664-654.
13. D. E. Knuth, "The Art of Computer Programming", Addison-Wesley Publishing Co.,, 1981.
14. K. Nyberg and R. A. Rueppel, "Message recovery for signature scheme based on the discrete logarithm problem", Advances in Cryptology-Proceedings of EUROCRYPT'94, 1994, pp. 182-193.
15. C. P. Schnorr, "Efficient signature generation for smart cards", Journal of Cryptology, Vol. 4(3), 1991, 239-252.
16. D. Song, "Practical Forward-Secure group signature schemes", In proceeding of 2001 ACM Symposium on Computer and Communication Security, 2001.