Journal of Information Processing Vol. 16 110–121 (Sep. 2008)

Regular Paper

Efficient Revocable Group Signature Schemes Using Primes

Toru Nakanishi[†1] and Nobuo Funabiki[†1]

Group signature schemes with membership revocation have been intensively researched. In this paper, we propose revocable group signature schemes with less computational costs for signing/verification than an existing scheme without the update of the signer's secret keys. The key idea is the use of a small prime number embedded in the signer's secret key. By using simple prime integer relations between the secret prime and the public product of primes for valid or invalid members, the revocation is efficiently ensured. To show the practicality, we implemented the schemes and measured the signing/verification times in a common PC (Core2 DUO 2.13 GHz). The times are all less than 0.5 seconds even for relatively large groups (10,000 members), and thus our schemes are sufficiently practical.

1. Introduction

1.1 Backgrounds

A *group signature scheme* [1),2),5),6),8),9),12),16),20),21)] allows a group member to anonymously sign a message on behalf of a group. In the group signature scheme, a group manager controls the membership of members, and the manager (or a third party) can cancel the anonymity of signatures to trace the signers. One of important topics in the group signature scheme is membership revocation [2),5),6),9),20),21)]. Namely, the membership of a member should be disabled without influencing the other members.

Some schemes [2),5),6),8),9),21)] deal with the membership revocation. However, in the schemes of Refs. 2), 6), the verification requires a computation with $O(R)$ complexity, where R is the number of revoked members.

In Ref. 9), an approach using a dynamic accumulator is proposed with the signing/verification costs of $O(1)$ w.r.t. N and R, where N is the group size. However, whenever making a signature, the signer has to modify a secret key. The computation of the modification is linear on the number of joining and removed members since the last time he signed. In the worst case, it is $O(N)$. In Ref. 8) using a similar approach, the signer's modification of his secret key requires the computation depending on the number of revocations since the last time he signed. Thus, in the worst case, the signer is required $O(R)$ computation. In Ref. 5), a group signature scheme from bilinear maps is proposed, where the same revocation approach is adopted. Therefore, since these schemes [5),8),9)] force a signer to modify his secret key with $O(N)$ or $O(R)$ complexity before signing, they are not suitable for large groups.

1.2 Previous Works

In Ref. 21), a revocable scheme suitable for middle-scale groups with about 1,000 members is proposed. For such groups, the signing/verification are almost independent from N and R, and any signer does not need to modify the secret key. However, for larger groups, the signing/verification requires a cost that depends on N/ℓ_n, where ℓ_n is a security parameter of strong RSA assumption, which is currently 1,024 or 2,048. In Ref. 20), an extended scheme suitable for the larger groups is proposed. In that scheme, the group is partitioned into sub-groups, and the scheme of Ref. 21) is utilized for each smaller sub-groups. In addition, it is kept secret which sub-group the signer belongs to.

Indeed, in the scheme of Ref. 21), the signing/verification algorithms have a complexity independent from N and R (w.r.t. the number of exponentiations), but the algorithms are complex due to the utilized zero-knowledge proofs. In this scheme, the manager publishes a membership data where each bit indicates whether the assigned member is valid or not. The group signature proves that the signer's bit is 1 while concealing which bit it is. In the proof, some integer equations and inequations on secret data are proved. However, since the zero-knowledge proof of the integer inequation based on Ref. 7) is complex, the group signature is also complex.

1.3 Our Contributions

In this paper, we propose revocable schemes with more efficient signing/verification algorithms. The first scheme is suitable for the small groups. In the scheme, a small and unique prime number is assigned to each member and is embedded in the member's secret key, and the manager publishes the product

†1 Okayama University

of primes of valid members. The group signature proves that the signer's prime is the factor of the published product. In the adopted zero-knowledge proof, the efficiency depends on the complexity of the proven relations and the sizes of the proven secrets. Since the relation of the product used in this scheme is very simple, the good efficiency is achieved. However, as the group size increases, the product of unique primes is also huge. Thus, since the zero-knowledge proof becomes inefficient due to the increase of the sizes of proven secrets, this scheme is not suitable for middle-scale or large groups.

In the second scheme, the manager publishes the product of primes of revoked members, and the group signature proves that the signer's prime is coprime to the published product. This relation is also simple. Generally, the number of revoked members is smaller than the number of valid members, and thus we can expect that this scheme will be more efficient than the first one.

Since the proven secrets of both schemes depend on N or R, these are not suitable for large groups. We furthermore apply the first scheme to the approach in Ref. 20) to obtain the extended scheme for large groups.

To show the effectiveness of the proposed schemes, we implemented the schemes on a PC. In this paper, we compare the execution times of the signing/verification with those in the scheme of Ref. 21).

1.4 Related Works

A scheme based on the idea similar to our second scheme is proposed in Ref. 13). In this scheme, a prime number is used for the revocation management, and the product of the primes of all revoked members is published. The group signature proves that the signer's prime is coprime to the published product. However, the serious problem is that the primes used in this scheme are large (about 1,000 bits), and thus the product is very huge. This is why the zero-knowledge proof based on the product is very inefficient even in the case where R is small. On the other hand, in our scheme, any prime number can be used. Thus, we can choose a unique prime from the smallest prime in sequence. Therefore, our second scheme is efficient even for the middle-scale group.

2. Model and Security Definitions

We show a model of group signature scheme with membership revocation. This model and the following security requirements are derived from Refs. 3), 4), 6), 18), except the *revocability*, which we introduce in this paper.

Group signature scheme with membership revocation consists of the following algorithms:

KeyGen: This probabilistic key generation algorithm for the group manager, on input 1^ℓ, outputs the group public key gpk, the initial membership data $\boldsymbol{md}[0]$ and manager's secret key msk.

Join: This is an interactive protocol between a probabilistic algorithm **Join-U** for the i-th user and a probabilistic algorithm **Join-GM** for the group manager, where the user joins the group controlled by the manager w.r.t. gpk. **Join-U**, on input gpk, outputs $\boldsymbol{usk}[i]$ that is the user's secret key. On the other hand, **Join-GM**, on inputs $gpk, msk, \boldsymbol{md}[t]$, outputs $\boldsymbol{reg}[i]$ and $\boldsymbol{md}[t+1]$.

Revoke: This probabilistic algorithm, on inputs gpk, $\boldsymbol{md}[t]$, revoked member's ID i, and $\boldsymbol{reg}[i]$, outputs new membership data $\boldsymbol{md}[t+1]$.

Sign: This probabilistic algorithm, on inputs gpk, $\boldsymbol{usk}[i]$, a membership data $\boldsymbol{md}[t]$, and signed message M, outputs the signature σ.

Verify: This is a deterministic algorithm for verification. The input is gpk, a membership data $\boldsymbol{md}[t]$, a signature σ, and the message M. Then the output is 'valid' or 'invalid'.

Open: This deterministic algorithm, on inputs gpk, msk, $\boldsymbol{reg}$, σ and M, outputs i, which indicates the signer of σ.

Then, the security requirements, *Traceability, Revocability, Anonymity*, and *Non-frameability*, are defined as follows.

2.1 Traceability

The following traceability requirement captures the unforgeability of group signatures. Consider the following traceability game between an adversary $\mathcal{A}$ and a challenger, where $\mathcal{A}$ tries to forge a signature that cannot be traced to one of members corrupted by $\mathcal{A}$.

Setup: The challenger runs **KeyGen**, and obtains gpk, msk and $\boldsymbol{md}[0]$. He provides $\mathcal{A}$ with gpk and $\boldsymbol{md}[0]$, and run $\mathcal{A}$. He sets $t = 0$, and sets CU and RCU with empty, where CU denotes the set of IDs of users corrupted by $\mathcal{A}$, and RCU denotes the set of IDs of revoked users in CU.

Queries: $\mathcal{A}$ can query the challenger about the following.

H-Join: $\mathcal{A}$ can request the i-th user's join. Then, the challenger executes the join protocol, where the challenger plays both the roles of the joining user and the manager. This corresponds to a honest user's join. Increase t by 1.

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 the manager. The challenger adds i to CU. This corresponds to a corrupted user's join. Increase t by 1.

Revocation: $\mathcal{A}$ requests the revocation of a member i. The challenger responds new $\boldsymbol{md}[t+1]$. The challenger adds i to RCU if $i \in CU$. Increase t by 1.

Signing: $\mathcal{A}$ requests a signature on a message M for a member i. The challenger responds the corresponding signature using the current $\boldsymbol{md}[t]$, 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.

Open: $\mathcal{A}$ requests to open a signature σ on the message M. The challenger responds the corresponding signer's ID i.

Output: Finally, $\mathcal{A}$ outputs a message M^* and a signature σ^*.

Then, $\mathcal{A}$ wins if

(1) $\mathbf{Verify}(gpk, \boldsymbol{md}[t], \sigma^*, M^*) = \text{valid}$,

(2) for $i^* = \mathbf{Open}(gpk, msk, \boldsymbol{reg}, \sigma^*, M^*)$, $i^* \notin CU$, and

(3) $\mathcal{A}$ did not obtain σ^* by making a signing query at M^*.

Traceability requires that for all PPT $\mathcal{A}$, the probability that $\mathcal{A}$ wins the traceability game is negligible.

2.2 Revocability

The revocability captures that any revoked member cannot compute any signature accepted by honest **Verify**. This property can be formalized in the framework of the traceability, as in Ref. 6). However, to clearly distinguish the meanings of the traceability and the revocability, we separately define these requirements.

Consider the revocability game that is the same game as the traceability except that $\mathcal{A}$ wins if

(1) $\mathbf{Verify}(gpk, \boldsymbol{md}[t], \sigma^*, M^*) = \text{valid}$,

(2) $CU = RCU$.

(3) $\mathcal{A}$ did not obtain σ^* by making a signing query at M^*.

In this game, $\mathcal{A}$ tries to forge a valid signature, in the situation that all corrupted members are revoked.

Revocability requires that for all PPT $\mathcal{A}$, the probability that $\mathcal{A}$ wins the revocability game is negligible.

2.3 Anonymity

The following anonymity requirement captures the anonymity and unlinkability of signatures. Consider the following anonymity game.

Setup: The challenger runs **KeyGen**, and obtains gpk, msk and $\boldsymbol{md}[0]$. He provides $\mathcal{A}$ with gpk and $\boldsymbol{md}[0]$, and runs $\mathcal{A}$. He sets $t = 0$ and CU with empty.

Queries: $\mathcal{A}$ can query the challenger. The available queries are the same ones as in the traceability game.

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 chooses $\phi \in_R \{0, 1\}$, and responds the signature on M of member i_ϕ using the current $\boldsymbol{md}[t]$.

Restricted queries: Similarly, $\mathcal{A}$ can make the queries. However, $\mathcal{A}$ cannot query the opening of the signature responded in the challenge.

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 $|\Pr[\phi' = \phi] - 1/2|$.

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

2.4 Non-frameability

This property requires that a signature of an honest member cannot be computed by other members and even the manager.

Consider the following non-frameability game.

Setup: The challenger runs **KeyGen**, and obtains gpk, msk and $\boldsymbol{md}[0]$. He provides $\mathcal{A}$ with gpk, msk and $\boldsymbol{md}[0]$, and run $\mathcal{A}$. He sets $t = 0$ and HU with empty, where HU denotes the set of IDs of honest users who are not corrupted by $\mathcal{A}$.

Queries: In the run, $\mathcal{A}$ issues the following queries to the challenger.

- **H-Join:** $\mathcal{A}$ can request the i-th honest user's join. Then, $\mathcal{A}$ as the manager executes the join protocol with the challenger as the i-the user. The challenger adds i to HU. Increase t by 1.
- **C-Join:** $\mathcal{A}$ can request the i-th corrupted user's join. $\mathcal{A}$ plays the roles of the manager and the i-th user. Thus, the challenger just obtains a new $\boldsymbol{md}[t+1]$. Increase t by 1.
- **Revocation:** $\mathcal{A}$ can request the revocation of member i. $\mathcal{A}$ plays the role of the manager to execute the revocation of the member. Thus, the challenger just obtains a new $\boldsymbol{md}[t+1]$. Increase t by 1.
- **Sign:** $\mathcal{A}$ can request a signature for a signed message M, user's ID i, and the current $\boldsymbol{md}[t]$. The challenger replies $\mathbf{Sign}(gpk, \boldsymbol{usk}[i], \boldsymbol{md}[t], M)$, if $i \in HU$.
- **Corruption:** $\mathcal{A}$ can request to corrupt a member by sending the member's ID i. The challenger returns $\boldsymbol{usk}[i]$, if $i \in HU$. The challenger deletes i from HU.

Output: Finally, $\mathcal{A}$ outputs a message M^* and a signature σ^*.
Then, $\mathcal{A}$ wins if

(1) $\mathbf{Verify}(gpk, \boldsymbol{md}[t], \sigma^*, M^*) =$ valid,
(2) for $i^* = \mathbf{Open}(gpk, msk, \boldsymbol{reg}, \sigma^*, M^*)$, $i^* \in HU$, and
(3) $\mathcal{A}$ did not obtain σ^* by making a signing query at M^*.

Non-Frameability requires that for all PPT $\mathcal{A}$, the probability that $\mathcal{A}$ wins the non-frameability game is negligible.

3. Preliminaries

3.1 Assumptions

We adopt the strong RSA assumption. Let $n = pq$ be an RSA modulus for safe primes p, q (i.e., $p = 2p' + 1, q = 2q' + 1$, and p, q, p', q' are prime), and let $QR(n)$ be the set of quadratic residues modulo n, that is, the cyclic subgroup of $\mathbb{Z}_n^*$ generated by an element of order $p'q'$. The strong RSA assumption on $QR(n)$ means that finding $(u \in QR(n), e \in \mathbb{Z}_{>1})$ s.t. $u^e = z \pmod n$ on inputs $(n, z \in QR(n))$ is infeasible. We furthermore adopt the DDH assumption.

3.2 CL Signature Scheme

Our group signature schemes utilize Camenisch-Lysyanskaya (CL) signature scheme [10]. The adopted version is the one in Ref. 8) with a slight modification.

Key generation: Let ℓ_n be a security parameter. The secret key consists of safe primes p, q, and the public key consists of $n = pq$ of length ℓ_n and $a, g_1, \ldots, g_L, h \in_R QR(n)$, where L is the number of blocks.

Signing: Let ℓ_m be a parameter. Given messages $m_1, \ldots, m_L \in \pm\{0,1\}^{\ell_m}$, choose a random number r of length ℓ_r (r can be chosen from $\mathbb{Z}_n$ in the applied group signature scheme [8]) and a random prime e of length ℓ_e s.t. $\ell_e \geq \ell_m + 2$. Compute y s.t. $y = (a g_1^{m_1} \cdots g_L^{m_L} h^r)^{1/e} \bmod n$. The signature is (e, y, r).

Verification: Given messages $m_1, \ldots, m_L$ and the signature (e, y, r), check $y^e = a g_1^{m_1} \cdots g_L^{m_L} h^r \pmod n$ and $2^{\ell_e - 1} < e < 2^{\ell_e}$.

3.3 *SPK*

As main building blocks, we adopt the signatures converted by Fiat-Shamir heuristic [15] from honest-verifier zero-knowledge proofs of knowledge, abbreviated as SPK. The adopted proofs show the relations among secret representations. The SPK of a representation on $QR(n)$ is proposed in Ref. 14). We furthermore use the SPK of representations with equal parts [11] (on different groups), and SPK of a representation with parts in intervals [11]. The details are described in Appendix. We adopt the notations on SPK in Refs. 8), 11). For instance, $SPK\{(\alpha, \beta, \gamma) : y = g^\alpha h^\beta \wedge \tilde{y} = \tilde{g}\tilde{h}^\gamma \wedge \alpha \in [u, v]\}(m)$ denotes the signature on message m proving the knowledge of (α, β, γ) s.t. $y = g^\alpha h^\beta$, $\tilde{y} = \tilde{g}\tilde{h}^\gamma$, and $\alpha \in [u, v]$. In the case where the message m is not needed, we omit (m) from this notation.

4. Basic Scheme

4.1 Idea

In the proposed scheme, we utilize the component of the scheme of Ref. 8) for the anonymous group membership authentication, and attach a revocation mechanism to it. The underlying scheme [8] is the most efficient among the RSA-based schemes.

The scheme is informally as follows. When a member joins, the member sends

$f(x)$ to the manager, where f is a one-way function and x is a secret. The manager returns a CL signature $S = Sign(x)$ to the member, where $Sign$ is the signing function of the manager in CL signature. Then, the group signature consists of $E = Enc(f(x))$, where Enc is a CCA2 secure ElGamal encryption function using the manager's public key, and the following SPK on the signed message M.

$$SPK\{(x, S) : S = Sign(x) \wedge E = Enc(f(x))\}(M).$$

When opening the signature, the manager decrypts E to check the sender of $f(x)$ in joining.

To obtain the revocability, the proposed scheme adopts a CL signature $S = Sign(x, d)$ for two messages, where d is a unique prime assigned to the joining member. To reduce the size of d, d is chosen as small as possible and different from other member's d. In addition, the manager publishes the product D of d of all valid members. The group signature consists of E and the following SPK.

$$SPK\{(x, d, S, k) : S = Sign(x, d) \wedge E = Enc(f(x)) \wedge D = kd\}(M).$$

A revoked member cannot prove $D = kd$ on his d embedded in S, and thus the revocability is ensured. Since this SPK excludes the SPK for integer inequations (which is used in the previous work [21], as mentioned in Introduction), the signing/verification is efficient unless D becomes huge (i.e., for small groups).

4.2 Proposed Algorithms

As well as the underlying scheme [8], our scheme has security parameters $\ell_n, \ell_m, \ell_e, \ell_s, \ell_c, \ell_E, \ell_Q, \ell_d$ s.t. $\ell_c + \ell_e + \ell_s + 1 < \ell_Q$ and $\max(\ell_Q, \ell_d) + \ell_c + \ell_s + 1 < \ell_E < \ell_n/2$. Based on the suggestion in Ref. 8), we adopt $\ell_n = 2{,}048, \ell_P = 1{,}600, \ell_Q = 282, \ell_c = 160, \ell_e = \ell_s = 60$ and variable ℓ_d, ℓ_E depending on N or R.

KeyGen:

The input is 1^{ℓ_n}, where ℓ_n is the security parameter.

(1) As the setup the CL signature scheme, compute two $(\ell_n/2)$-bit safe primes p, q and $n = pq$, and chooses $a, g_1, g_2, h \in_R QR(n)$.

(2) As the setup of the CCA2 secure ElGamal cryptosystem, choose ℓ_Q-bit and ℓ_P-bit primes Q, P s.t. $Q|P-1$. Let F be an element of order Q in $\mathbb{Z}_P^*$. Choose $X_G, X_H \in_R \mathbb{Z}_Q$ and compute $G = F^{X_G} \bmod P$ and $H = F^{X_H} \bmod P$.

(3) Output $gpk = (n, a, g_1, g_2, h, Q, P, F, G, H)$, $msk = (p, q, X_G)$, and $\boldsymbol{md}[0] = D_0$, where $D_0 = 1$.

Join:

This interactive protocol is conducted on an authenticated channel. The common inputs of **Join-U** and **Join-GM** is $gpk = (n, a, g_1, g_2, h, Q, P, F, G, H)$. The input of **Join-GM** consists of $msk = (p, q, X_G)$ and $\boldsymbol{md}[t] = D_t$.

(1) [**Join-U**] Choose $x_i \in_R \mathbb{Z}_Q$, and generate $Y_i = G^{x_i} \bmod P$. This corresponds to $f(x)$ explained in Section 4.1. Then, choose $r_i' \in_R \mathbb{Z}_n$, and compute $c_i = g_1^{x_i} h^{r_i'} \bmod n$. Send Y_i, c_i to **Join-GM**. In addition, compute

$$W_i = SPK\{(\xi, \rho') : c_i = g_1^{\xi} h^{\rho'} \bmod n \wedge Y_i = G^{\xi} \bmod P\},$$

and send **Join-GM** it.

(2) [**Join-GM**] If this SPK is invalid, abort. Otherwise, choose $e_i \in_R \{0,1\}^{\ell_e}$ s.t. $E_i = 2^{\ell_E} + e_i$ is prime. Assign a unique prime d_i to this member i, where d_i is chosen as small as possible and different from other member's d_i. Choose $r_i'' \in_R \{0,1\}^{\ell_e}$, and set $y_i = (a g_1^{x_i} g_2^{d_i} h^{r_i' + r_i''})^{E_i^{-1}}$. Send (y_i, E_i, r_i'') back to **Join-U**. Compute $D_{t+1} = D_t d_i$. Output $\boldsymbol{reg}[i] = (Y_i, d_i)$ and $\boldsymbol{md}[t+1] = D_{t+1}$.

(3) [**Join-U**] Output $\boldsymbol{usk}[i] = (x_i, d_i, r_i = r_i' + r_i'', y_i, e_i = E_i - 2^{\ell_E})$. The CL signature (E_i, y_i, r_i) of messages x_i, d_i corresponds to $S = Sign(x, d)$ explained in Section 4.1.

Revoke:

The inputs are gpk, $\boldsymbol{md}[t] = D_t$, revoked member's ID i, and $\boldsymbol{reg}[i] = (Y_i, d_i)$. Compute $D_{t+1} = D_t / d_i$. Output $\boldsymbol{md}[t+1] = D_{t+1}$.

Sign:

The inputs of this signing algorithm are $gpk = (n, a, g_1, g_2, h, Q, P, F, G, H)$, the signer's secret $\boldsymbol{usk}[i] = (x_i, d_i, r_i, y_i, e_i)$, the membership data $\boldsymbol{md}[t] = D_t$, and a signed message $M \in \{0,1\}^*$. The algorithm is as follows:

(1) Select $r \in_R \{0,1\}^{\ell_n/2}$ and $R \in_R \mathbb{Z}_Q$. Set $u = h^r y_i \bmod n$, $U_1 = F^R \bmod P$, $U_2 = G^R Y_i \bmod P$, and $U_3 = H^{R+e_i} \bmod P$. (U_1, U_2, U_3) corresponds to $E = Enc(f(x))$ explained in Section 4.1.

(2) Compute

$$V = SPK\{(\xi, \delta, \rho, \epsilon, \alpha, \beta, \gamma, \zeta, \tau) : a = u^{2^{\ell_E+\epsilon}} g_1^{-\xi} g_2^{-\delta} h^{\rho} \bmod n$$
$$\wedge g_2^{D_t} = a^{-\alpha} u^{\beta} g_1^{-\gamma} h^{\zeta} \bmod n$$
$$\wedge U_1 = F^{\tau} \wedge U_2 = G^{\tau+\xi} \wedge U_3 = H^{\tau+\epsilon}$$
$$\wedge \epsilon \in \{-2^{\ell_e+\ell_c+\ell_s}, 2^{\ell_e+\ell_c+\ell_s}\} \wedge \xi \in \{-2^{\ell_Q+\ell_c+\ell_s}, 2^{\ell_Q+\ell_c+\ell_s}\}$$
$$\wedge \delta \in \{-2^{\ell_d+\ell_c+\ell_s}, 2^{\ell_d+\ell_c+\ell_s}\}\}(M).$$

This SPK corresponds to

$$SPK\{(x, d, S, k) : S = Sign(x, d) \wedge E = Enc(f(x)) \wedge D = kd\}(M),$$

explained in Section 4.1. In the SPK V, the first equation and the last three range relations correspond to $S = Sign(x, d)$, the second equation corresponds to $D = kd$, and the remaining equations including U_1, U_2, U_3 corresponds to $E = Enc(f(x))$.

(3) Output the group signature $\sigma = (u, U_1, U_2, U_3, V)$.

Verify:

The inputs are gpk, $\boldsymbol{md}[t]$, a target signature $\sigma = (u, U_1, U_2, U_3, V)$ and the message M. Check the SPK V. Output 'valid' (resp., 'invalid') if it is correct (resp., incorrect).

Open:

The inputs are gpk, the secret key $msk = (p, q, X_G)$, $\boldsymbol{reg}$ with $\boldsymbol{reg}[i] = (Y_i, d_i)$, a target signature $\sigma = (u, U_1, U_2, U_3, V)$ and the message M.

(1) Verify σ. If it is invalid, abort.

(2) Using X_G, decrypt (U_1, U_2) to obtain G^{x_i}. Search $\boldsymbol{reg}$ for i with $Y_i = G^{x_i}$.

(3) Output i.

4.3 Security

Theorem 1 The basic scheme has the traceability under the strong RSA assumption in the random oracle model.

Proof sketch.

Assume an adversary for the traceability game, and we will build an adversary for CL signature. Let **CL-SO** be the signing oracle for CL signature. Consider the following framework with the adversary $\mathcal{A}$ for the traceability game. Let N be the maximum group size.

Setup. It is given the public key (n, a, g_1, g_2, h) of CL signature. As the real algorithm, generate X_G, X_H and (Q, P, F, G, H). As the public key of group signature, $gpk = (n, a, g_1, g_2, h, Q, P, F, G, H)$ is given to $\mathcal{A}$. Select $i^* \in_R [1, N]$.

Queries. Each query is simulated as follows.

Hash queries. At any time, $\mathcal{A}$ can query the hash function used in SPKs. Responds with random values with consistency.

H-Join: Execute the following, according to i.

Case of $i \neq i^*$: As **Join-U**, execute the real algorithm to produce (x_i, Y_i, c_i). In **Join-GM**, choose d_i as usual. Then, send messages (x_i, d_i) to **CL-SO**, which returns the CL signature (r_i, y_i, E_i) s.t. $y_i = (a g_1^{x_i} g_2^{d_i} h^{r_i})^{E_i^{-1}} \bmod n$. Generate D_{t+1} as usual.

Case of $i = i^*$: As usual, select $x_{i^*}, d_{i^*}, e_{i^*}$, and compute Y_{i^*} and renew D_{t+1}. Note that, in this case, the CL signature is not obtained, i.e., **CL-SO** is not queried for i^*.

C-Join: $\mathcal{A}$ that plays **Join-U** sends (Y_i, c_i, W_i). Using the extractor of the SPK W_i, obtain x_i, r_i'. Choose d_i as usual. Then, send the messages (x_i, d_i) to **CL-SO**, which returns the CL signature (r_i, y_i, E_i) s.t. $y_i = (a g_1^{x_i} g_2^{d_i} h^{r_i})^{E_i^{-1}} \bmod n$. Set $r_i'' = r_i - r_i'$, and return (y_i, E_i, r_i'') to $\mathcal{A}$. Generate D_{t+1} as usual.

Revocation: This is the same as the real algorithm.

Signing: Execute the following, according to i.

Case of $i \neq i^*$: Since we know x_i via **H-Join** together with (d_i, r_i, y_i, e_i), execute the real algorithm.

Case of $i = i^*$: In this case, we does not know $\boldsymbol{usk}[i^*]$, and simulate the signature of i^* as follows. Select $r \in_R \{0,1\}^{\ell_n/2}$ and $R \in_R \mathbb{Z}_Q$. Set $u = h^r \bmod n$, $U_1 = F^R \bmod P$, $U_2 = G^R Y_i \bmod P$, and $U_3 = H^{R+e_i} \bmod P$. The SPK V is simulated by the zero-knowledge simulator.

Corruption: If i^* is requested, abort. Otherwise, return the requested secret key $\boldsymbol{usk}[i] = (x_i, d_i, r_i, y_i, e_i)$ produced in **H-Join**.

Open: This is the same as the real, using X_G.

Output: With a non-negligible probability, $\mathcal{A}$ outputs a message M^* and the signature σ^* satisfying the winning conditions.

Since σ^* is a valid signature, by using the extractor of the SPK V, we can

obtain $(e, x, d, \tilde{r})$ s.t. $a = u^{2^{\ell_E}+e} g_1^{-x} g_2^{-d} h^{\tilde{r}} \bmod n$. This means the existence of a CL signature for messages x, d. From the winning condition on **Open**, $x = x_i$ for i joining by **H-Join**, or x is different from all x_i. In the former, $x = x_{i^*}$ holds with a non-negligible probability, since the selection of i^* is independent from the view of $\mathcal{A}$. In both cases, no signature for x is requested to **CL-SO**. This means that CL signature scheme is broken, which contradicts the strong RSA assumption. Therefore, the traceability holds under the assumption. □

Theorem 2 The basic scheme has the revocability under the strong RSA assumption in the random oracle model.

Proof sketch.

Assume $\mathcal{A}$ for the revocability game, and we will construct the adversary for CL signature. The used framework is the same as in the case of the traceability. From the winning conditions of the revocability game, member i joining by **C-Join** and corrupted member i are all revoked before the output. Finally, $\mathcal{A}$ outputs M^* and a valid σ^* with a non-negligible probability. Then, using the extractor of the SPK V, we can extract $(e, x, d, \tilde{r})$ s.t. $a = u^{2^{\ell_E}+e} g_1^{-x} g_2^{-d} h^{\tilde{r}} \bmod n$ and $g_2^{D_t} = a^{-\alpha} u^{\beta} g_1^{-\gamma} h^{\zeta} \bmod n$. From both equations,

$$g_2^{D_t} = (u^{2^{\ell_E}+e_i} g_1^{-x} g_2^{-d} h^{\tilde{r}})^{-\alpha} u^{\beta} g_1^{-\gamma} h^{\zeta} \bmod n,$$
$$1 = u^{-(2^{\ell_E}+e_i)\alpha+\beta} g_1^{x\alpha-\gamma} g_2^{d\alpha-D_t} h^{-\tilde{r}\alpha+\zeta} \bmod n.$$

Thus, from the strong RSA assumption, we obtain $d\alpha - D_t = 0$, i.e., $D_t = d\alpha$, in $\mathbb{Z}$. This means that d is a prime factor of D_t or a product of the factors. Assume that d is a product of the prime factors of D_t. Then, such a d has never been sent to **CL-SO**, and thus CL signature scheme is broken. Thus, consider only the case that d is a prime factor of D_t. This implies that d is for a member that was honestly joined by **H-join** or **C-join** but is not revoked. However, since members joined by **C-join** were revoked, the member with d has to be joined by **H-join**. By the same discussion as in Theorem 1, with a non-negligible probability, a CL signature extracted from σ^* is not requested to **CL-SO**, and thus CL signature scheme is broken. Therefore, the revocability holds under the strong RSA assumption. □

Theorem 3 The basic scheme has the anonymity under the DDH assumption in the random oracle model.

This proof is the same as the underlying scheme [8], since our signature has the same structure as that of Ref. 8), i.e., a CCA2 secure encryption of the identity of the signer and the SPK.

Theorem 4 The basic scheme has the non-frameability under the DL assumption in the random oracle model.

Proof sketch.

We will construct an adversary for the DL assumption, using the adversary $\mathcal{A}$ for the non-frameability game.

The input for the DL assumption is (P, Q, G, G^a) for G with order $Q \in \mathbb{Z}_P^*$ and $a \in \mathbb{Z}_Q$. Choose $i^* \in_R [1, N]$ as the above proofs. Then, run the real **KeyGen** except that the given P, Q, G are used and $F = G^{1/X_G} \bmod P$. Provide $\mathcal{A}$ with gpk, msk and $\boldsymbol{md}[0]$, and run $\mathcal{A}$. In the run, the queries for $i \neq i^*$ are replied by the real algorithms. In case of $i = i^*$, define $x_{i^*} = a$ that is unknown for the challenger, and the other parameters are the same as in the real algorithms. Then, we need the simulations of $Y_{i^*}, c_{i^*}, W_{i^*}$ in **H-Join** and u, U_2, V in **Signing** without the knowledge of $x_{i^*} = a$. Since c_{i^*}, u are commitments, they are simulated by randoms. Since W_{i^*}, V are the SPK, they are simulated by the zero-knowledge simulators. $Y_{i^*} = G^{x_{i^*}} \bmod P$ can be simulated by the given G^a, and U_2 is similar. Thus, by the similar discussion to the proof in the traceability, with a non-negligible probability, $\mathcal{A}$ outputs a signature σ^* of the target i^*. Then, we can extract x_{i^*} from the SPK V by the extractor. Since $x_{i^*} = a$, this means that the DL assumption is broken. Therefore, the non-frameability holds under the DL assumption. □

5. Extended Scheme for Middle-Scale Groups

5.1 Idea

In the basic scheme, as the group grows, the signing/verification becomes inefficient, since the product D of all valid primes becomes huge. Generally, the number of revoked members R is much less than the number of valid members. Here, we extend the basic scheme to a one where D is the product of all primes for only revoked members. This scheme is expected to be suitable for middle-scale groups.

The construction of this scheme is similar to that of the basic scheme. The joining member is issued a CL signature $S = Sign(x, d)$ for a unique prime d. On the other hand, the manager publishes the product D of d of all revoked members. Then, the group signature consists of E and the following SPK.

$$SPK\{(x, d, S, k_1, k_2) : S = Sign(x, d) \wedge E = Enc(f(x)) \\ \wedge k_1 D + k_2 d = 1\}(M).$$

$k_1 D + k_2 d = 1$ means that d is coprime to D, i.e., the member with d is not revoked.

5.2 Proposed Algorithms

Algorithms **KeyGen** and **Open** are the same as those in the basic scheme.

Join:

This is the same as that in the basic scheme, except that D is not changed, i.e., $D_{t+1} = D_t$, in Step 2.

Revoke:

Compute $D_{t+1} = D_t d_i$. Output $\boldsymbol{md}[t+1] = D_{t+1}$.

Sign, Verify:

From the basic scheme, the SPK is modified as follows.

$$V = SPK\{(\xi, \delta, \rho, \epsilon, \alpha, \beta, \gamma, \zeta, \eta, \tau) : a = u^{2^{\ell_E+\epsilon}} g_1^{-\xi} g_2^{-\delta} h^{\rho} \bmod n \\ \wedge g_2 = a^{-\alpha} u^{\beta} g_1^{-\gamma} (g_2^{D_t})^{\eta} h^{\zeta} \bmod n \\ \wedge U_1 = F^{\tau} \wedge U_2 = G^{\tau+\xi} \wedge U_3 = H^{\tau+\epsilon} \\ \wedge \epsilon \in \{-2^{\ell_e+\ell_c+\ell_s}, 2^{\ell_e+\ell_c+\ell_s}\} \wedge \xi \in \{-2^{\ell_Q+\ell_c+\ell_s}, 2^{\ell_Q+\ell_c+\ell_s}\} \\ \wedge \delta \in \{-2^{\ell_d+\ell_c+\ell_s}, 2^{\ell_d+\ell_c+\ell_s}\}\}(M).$$

5.3 Security

The security proof is similar to that of the basic scheme. In addition, in the proof of Theorem 2, we need to prove that the SPK proves the knowledge of α, η s.t. $d\alpha + D_t\eta = 1$ for the knowledge d extracted in the proof of Theorem 2 and $\boldsymbol{md}[t] = D_t$.

The relation is obtained as follows. From the above SPK, we can extract values s.t.

$$a = u^{2^{\ell_E+\epsilon}} g_1^{-\xi} g_2^{-\delta} h^{\rho} \bmod n, \quad g_2 = a^{-\alpha} u^{\beta} g_1^{-\gamma} (g_2^{D_t})^{\eta} h^{\zeta} \bmod n$$

Thus, we obtain

$$g_2 = (u^{2^{\ell_E+\epsilon}} g_1^{-\xi} g_2^{-\delta} h^{\rho})^{-\alpha} u^{\beta} g_1^{-\gamma} (g_2^{D_t})^{\eta} h^{\zeta} \bmod n \\ = u^{-2^{\ell_E+\epsilon}\alpha+\beta} g_1^{\xi\alpha-\gamma} g_2^{\delta\alpha+D_t\eta} h^{-\rho\alpha+\zeta} \bmod n.$$

Thus, we can obtain the relation $\delta\alpha + D_t\eta = 1$, for $\delta = d$ in the proof.

6. Extended Scheme for Large Groups

For larger groups, the previous extended scheme also suffers from the increase of R (and N due to the size of d_i). Here, we show an extended scheme from the basic scheme by the approach in Ref. 20). As the details on the approach are available in Ref. 20), we provide only the informal description here.

In the approach, the large group of members is partitioned into subgroups. In each subgroup, our revocation mechanism is used. Namely, the manager computes the product $D_t^{(j)}$ of valid members' primes for each subgroup j. Then, we can limit the size of $D_t^{(j)}$ to a constant. On the other hand, if $D_t^{(j)}$ is public in the group signature, the verifier can identify the subgroup that the signer belongs to. Thus, the manager publishes a CL signature for $D_t^{(j)}$ at the current time t to ensure the correctness, and the group signature proves only the knowledge of the signature and $D_t^{(j)}$. This allows the verifier to be convinced of the correctness of used $D_t^{(j)}$ while $D_t^{(j)}$ is kept secret.

The construction is informally as follows. In a join, the joining member is issued a CL signature $Sign(x_i, d_i, j)$, where j means the j-th subgroup that the member joins. The manager publishes $D_{t+1}^{(j)} = D_t^{(j)} d_i$ for the subgroup j ($D_{t+1}^{(\hat{j})} = D_t^{(\hat{j})}$ for others $\hat{j}$), and the CL signatures $Sign(D_{t+1}^{(\hat{j})}, \hat{j}, t+1)$ for all subgroups $\hat{j}$ at the current time $t+1$. In the revocation of member i in subgroup j, the manager publishes $D_{t+1}^{(j)} = D_t^{(j)}/d_i$ for the subgroup j ($D_{t+1}^{(\hat{j})} = D_t^{(\hat{j})}$ for others $\hat{j}$), and $Sign(D_{t+1}^{(\hat{j})}, \hat{j}, t+1)$ for all $\hat{j}$.

The group signature consists of E and the following SPK.

$$SPK\{(x_i, d_i, j, S_i, k, D_t^{(j)}, \tilde{S}_j) : S_i = Sign(x_i, d_i, j) \wedge E = Enc(f(x_i)) \\ \wedge D_t^{(j)} = k d_i \wedge \tilde{S}_j = Sign(D_t^{(j)}, j, t)\}(M),$$

where the current time t is public. This SPK ensures the correctness of $D_t^{(j)}$ via the CL signature while j is kept secret, in addition to the SPK of the basic

Table 1 Implementation environment.

CPU	Intel Core2 Duo 2.13 GHz
Memory	2 GB
OS	Linux (kernel 2.6.16)
Language	C (GMP 4.1.4)

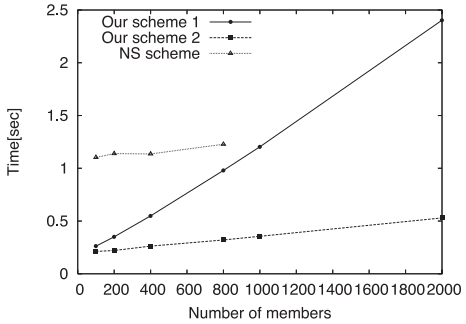

Fig. 1 Comparison of signing times.

scheme.

Since the size of $D_t^{(j)}$ is limited, the computational costs of the SPK, and thus the signing/verification are constant. The compensation is the overhead cost of the manager's generating $Sign(D_{t+1}^{(\hat{j})}, \hat{j}, t+1)$ for all $\hat{j}$, which is evaluated later.

7. Performance Measurements

To show the effectiveness of the proposed schemes, we implemented the schemes by using the GMP library [17], and measured the performances on a PC in the environment of **Table 1**. In the measurements, we use the setting of $\ell_n = 2{,}048, \ell_P = 1{,}600, \ell_Q = 282, \ell_c = 160, \ell_e = \ell_s = 60$ and we changed ℓ_d, ℓ_E depending on N or R.

We show the signing times in the proposed basic scheme (Our scheme 1), the extended scheme for middle-scale groups (Our scheme 2), and the scheme of Ref. 21) (NS scheme) in **Fig. 1**, and the verification times in **Fig. 2**. In our scheme 2, we assume that 10% of all members are revoked. NS scheme is adjusted by using the most efficient scheme of Ref. 8) as the base component, since we desire to compare the efficiency of the revocation mechanism. Since the combination of Refs. 21) and 8) limits the total number of members to about 800, we measured the time for at most 800 members for NS scheme. The figures show that our scheme 1 is more efficient than NS scheme for small groups (less than 500 members). The figures also show that our scheme 2 is much more efficient than scheme 1 in the likely situation that 10% of members are revoked, and efficient even for middle-scale groups (thousands members). However, this scheme becomes inefficient for larger groups or in the situation that the greater part of members are revoked.

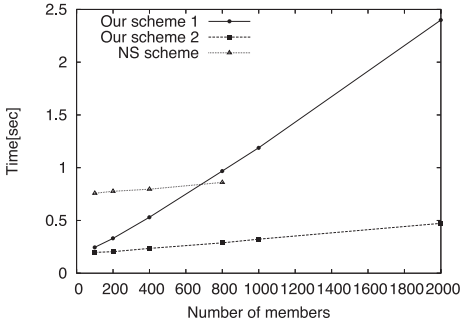

Fig. 2 Comparison of verification times.

Furthermore, we measured the times in the extended scheme for large groups in Section 6, where we set the size of the subgroups as 100. The signing time is 0.41 seconds with no dependency on N or R, and the verification time is 0.36 seconds. Note that the manager has an overhead cost for generating CL signature $Sign(D_t^{(j)}, j, t)$ for all subgroups in each join and revocation. For example, in case of $N = 10{,}000$, the overhead cost is 100 generations. However, this can be treated by powerful servers, and thus this is not a severe problem. One problem for large or huge groups is the massive amount of public data, $D_t^{(j)}$ and $Sign(D_t^{(j)}, j, t)$ for all j. In case that the subgroup size is 100, each $D_t^{(j)}$ requires 730 bits (since the product of the first 100 primes 2, 3, ..., 541 is 730 bits), and $Sign(D_t^{(j)}, j, t)$ requires 4156 bits (since e, r, y are 60, 2,048, 2,048 bits respectively in the CL signature). Thus, in case of $N = 100{,}000$, since the number of subgroups is 1,000, the public data run up to about 600 KB. Therefore, although our scheme becomes impractical for such huge groups, it is sufficiently applicable

for large groups (up to about 10,000 members), due to the good performance of the signing/verification.

8. Conclusion

We proposed efficient group signature schemes with revocation using prime relations. From the measurements in the implementation, the signing/verification of the basic scheme is more efficient than the previous scheme [21] for small groups. Furthermore, we have shown that the extended schemes have practical efficiency for middle-scale and large groups.

One of our future work is to apply the proposed schemes to the user authentication such as SSL/TLS.

Acknowledgments We would like to thank Naoto Hamada and Takuya Nakayama for helpful works.

References

1) Ateniese, G., Camenisch, J., Joye, M. and Tsudik, G.: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme, *Advances in Cryptology — CRYPTO 2000*, LNCS 1880, pp.255–270, Springer-Verlag (2000).
2) Ateniese, G., Song, D. and Tsudik, G.: Quasi-Efficient Revocation of Group Signatures, *Proc. 6th Financial Cryptography Conference* (*FC 2002*), LNCS 2357, pp.183–197, Springer-Verlag (2003).
3) Bellare, M., Shi, H. and Zhang, C.: Foundations of Group Signatures: The Case of Dynamic Groups, *Topics in Cryptology — CT-RSA 2005*, LNCS 3376, pp.136–153, Springer-Verlag (2005).
4) Bellare, M., Micciancio, D. and Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction based on General Assumptions, *Advances in Cryptology — EUROCRYPT 2003*, LNCS 2656, pp.614–629, Springer-Verlag (2003).
5) Boneh, D., Boyen, X. and Shacham, H.: Short Group Signatures, *Advances in Cryptology — CRYPTO 2004*, LNCS 3152, pp.41–55, Springer-Verlag (2004).
6) Boneh, D. and Shacham, H.: Group Signatures with Verifier-Local Revocation, *Proc. 11th ACM Conference on Computer and Communications Security* (*ACM-CCS '04*), pp.168–177 (2004).
7) Boudot, F.: Efficient proofs that a committed number lies in an interval, *Advances in Cryptology — EUROCRYPT 2000*, LNCS 1807, pp.431–444, Springer-Verlag (2000).
8) Camenisch, J. and Groth, J.: Group Signatures: Better Efficiency and New Theoretical Aspects, *Security in Communication Networks: 4th International Conference, SCN 2004*, LNCS 3352, pp.120–133, Springer-Verlag (2005).
9) Camenisch, J. and Lysyanskaya, A.: Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials, *Advances in Cryptology — CRYPTO 2002*, LNCS 2442, pp.61–76, Springer-Verlag (2002).
10) Camenisch, J. and Lysyanskaya, A.: A Signature Scheme with Efficient Protocols, *Proc. 3rd Conference on Security in Communication Networks* (*SCN '02*), LNCS 2576, pp.268–289, Springer-Verlag (2002).
11) Camenisch, J. and Michels, M.: Separability and Efficiency for Generic Group Signature Schemes, *Advances in Cryptology — CRYPTO '99*, LNCS 1666, pp.413–430, Springer-Verlag (1999).
12) Chaum, D. and van Heijst, E.: Group Signatures, *Advances in Cryptology — EUROCRYPT '91*, LNCS 547, pp.241–246, Springer-Verlag (1991).
13) Chen, Z., Wang, J., Wang, Y., Huang, J. and Huang, D.: An Efficient Revocation Algorithm in Group Signatures, *Proc. 6th International Conference on Information Security and Cryptology* (*ICISC2003*), LNCS 2971, pp.339–351, Springer-Verlag (2004).
14) Damgård, I. and Fujisaki, E.: A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order, *Advances in Cryptology — ASIACRYPT 2002*, LNCS 2501, pp.125–142, Springer-Verlag (2002).
15) Fiat, A. and Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems, *Advances in Cryptology — CRYPTO '86*, LNCS 263, pp.186–194, Springer-Verlag (1987).
16) Furukawa, J. and Imai, H.: An Efficient Group Signature Scheme from Bilinear Maps, *Proc. 10th Australasian Conference on Information Security and Privacy* (*ACISP 2005*), LNCS 3574, pp.455–467, Springer-Verlag (2005).
17) GNU Multiple Precision Arithmetic Library (GMP). http://gmplib.org/
18) Kiayias, A., Tsiounis, Y. and Yung, M.: Traceable Signatures, *Advances in Cryptology — EUROCRYPT 2004*, LNCS 3027, pp.571–589, Springer-Verlag (2004).
19) Lysyanskaya, A.: Signature Schemes and Applications to Cryptographic Protocol Design, Ph.D. Thesis, Massachusetts Institute of Technology (2002). Available in http://www.cs.brown.edu/~anna/phd.ps
20) Nakanishi, T., Kubooka, F., Hamada, N. and Funabiki, N.: Group Signature Schemes with Membership Revocation for Large Groups, *Proc. 10th Australasian Conference on Information Security and Privacy* (*ACISP 2005*), LNCS 3574, pp.443–454, Springer-Verlag (2005).
21) Nakanishi, T. and Sugiyama, Y.: A Group Signature Scheme with Efficient Membership Revocation for Reasonable Groups, *Proc. 9th Australasian Conference on Information Security and Privacy* (*ACISP 2004*), LNCS 3108, pp.336–347, Springer-Verlag (2004).

Appendix

A.1 Details of *SPK*s

We review the detail of the primitive SPKs. Let $\mathcal{H}$ be a collision-resistant hash function such that $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{\ell_c}$, for a security parameter ℓ_c. We also use a parameter ℓ_s. Hereafter, we use the notations: Let $[a, a+d]$ be the integer interval of all integers int such that $a \leq int \leq a+d$, for an integer a and a positive integer d. Let $[a, a+d)$ be the integer interval of all int such that $a \leq int < a+d$, and let $(a, a+d)$ be the interval for all int such that $a < int < a+d$. $S_1 \| S_2$ indicates the concatenation of S_1 and S_2 as bit strings.

A.1.1 *SPK* of Representation

Here, we introduce a version in the paper due to Lysyanskaya [19]. Hereafter, though the case of two bases $g, h \in QR(n)$ is described for all the SPKs, it is easy to generalize to the case of more bases.

Let $C = g^x h^y$ for $x \in (-2^{\ell_x}, 2^{\ell_x})$ and $y \in (-2^{\ell_y}, 2^{\ell_y})$. Then, $SPK\{(\alpha, \beta) : C = g^\alpha h^\beta\}(m)$ is computed as follows: Choose $r_x \in_R [0, 2^{\ell_x+\ell_c+\ell_s})$ and $r_y \in_R [0, 2^{\ell_y+\ell_c+\ell_s})$, and compute $\tilde{C} = g^{r_x} h^{r_y}$. Then, set challenge $c_0 = \mathcal{H}(m \| g \| C \| \tilde{C})$, and compute responses $s_x = r_x + c_0 x$ and $s_y = r_y + c_0 y$ (both in $\mathbb{Z}$). The signature is (c_0, s_x, s_y). On the other hand, the verification is to check if $c_0 = \mathcal{H}(m \| g \| C \| C^{-c_0} g^{s_x} h^{s_y})$. The following lemma is derived from Ref. 19).

Lemma 1 Under the strong RSA assumption, the interactive version of the above construction is an honest-verifier zero-knowledge proof of knowledge of α, β.

A.1.2 *SPK* of Representations with Equal Parts

It is easy to obtain this SPK by adopting the same randomness r_x or r_y for the same knowledge in the SPKs for representations.

Let $C = g^x h^y$ and $C' = g^x h^z$ for $x \in (-2^{\ell_x}, 2^{\ell_x})$, $y \in (-2^{\ell_y}, 2^{\ell_y})$, and $z \in (-2^{\ell_z}, 2^{\ell_z})$. Then, $SPK\{(\alpha, \beta, \gamma) : C = g^\alpha h^\beta \wedge C' = g^\alpha h^\gamma\}(m)$ is computed as follows: Choose $r_x \in_R [0, 2^{\ell_x+\ell_c+\ell_s})$, $r_y \in_R [0, 2^{\ell_y+\ell_c+\ell_s})$, and $r_z \in_R [0, 2^{\ell_z+\ell_c+\ell_s})$, and compute $\tilde{C} = g^{r_x} h^{r_y}$ and $\tilde{C}' = g^{r_x} h^{r_z}$. Then, set $c_0 = \mathcal{H}(m \| g \| C \| C' \| \tilde{C} \| \tilde{C}')$, and compute $s_x = r_x + c_0 x$, $s_y = r_y + c_0 y$, and $s_z = r_z + c_0 z$ (in $\mathbb{Z}$). The signature is (c_0, s_x, s_y, s_z). On the other hand, the verification is to check if $c_0 = \mathcal{H}(m \| g \| C \| C' \| C^{-c_0} g^{s_x} h^{s_y} \| C'^{-c_0} g^{s_x} h^{s_z})$. The security can be proved in the similar way to the normal SPK for a representation.

Furthermore, This SPK can be applied to one between representations on $QR(n)$ and a group with a known prime order, such as the group generated by F (an element with order Q in $\mathbb{Z}_P^*$) in the proposed scheme [11].

A.1.3 *SPK* of Representation with Parts in Intervals

The SPK of a representation with parts that lie in expanded intervals, i.e., the proved interval is expanded from the interval in which the part lies in fact, is obtained from the normal SPK of a representation by adding the verification of the domain of the response s_x or s_y.

Let $C = g^x h^y$ for $x \in [a, a+d]$ and $y \in (-2^{\ell_y}, 2^{\ell_y})$, where a is an integer and d is a positive integer. Then, $SPK\{(\alpha, \beta) : C = g^\alpha h^\beta \wedge \alpha \in [a - 2^{\ell_c+\ell_s} d, a + 2^{\ell_c+\ell_s} d]\}(m)$ is computed as follows: Choose $r_x \in_R [0, 2^{\ell_c+\ell_s} d)$ and $r_y \in_R [0, 2^{\ell_y+\ell_c+\ell_s})$, and compute $\tilde{C} = g^{r_x} h^{r_y}$. Then, set $c_0 = \mathcal{H}(m \| g \| C \| \tilde{C})$, and compute $s_x = r_x + c_0(x - a)$ and $s_y = r_y + c_0 y$ (both in $\mathbb{Z}$). But, if $s_x \notin [c_0 d, 2^{\ell_c+\ell_s} d)$, start again. The signature is (c_0, s_x, s_y). On the other hand, the verification is to check if $c_0 = \mathcal{H}(m \| g \| C \| C^{-c_0} g^{s_x + c_0 a} h^{s_y})$ and $s_x \in [c_0 d, 2^{\ell_c+\ell_s} d)$. This convinces the verifier that $\alpha \in [a - 2^{\ell_c+\ell_s} d, a + 2^{\ell_c+\ell_s} d]$ that is expanded from the real interval $[a, a+d]$.

The security can be proved in the similar way to the normal SPK for a representation, except that the knowledge extracted by the knowledge extractor surely lies in the interval $[a - 2^{\ell_c+\ell_s} d, a + 2^{\ell_c+\ell_s} d]$. Note that the cost of this SPK is comparable with the normal SPK of a representation.

(Received November 27, 2007)
(Accepted June 3, 2008)
(Released September 10, 2008)

Toru Nakanishi received the 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.

Nobuo Funabiki received the 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 the 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 computer network, optimization algorithm, image processing, educational technology, Web technology, and network security. He is a member of IEEE and IEICE.