# Revocable Group Signatures with Compact Revocation List Using Accumulators

Toru Nakanishi(✉) and Nobuo Funabiki

Department of Communication Network Engineering, Okayama University, Okayama City, Japan
{nakanisi,funabiki}@cne.okayama-u.ac.jp

**Abstract.** Group signatures allow a group member to anonymously sign a message on behalf of the group. One of the important issues is the revocation, and lots of revocable schemes have been proposed so far. The scheme recently proposed by Libert et al. achieves that $O(1)$ or $O(\log N)$ efficiency except for the revocation list size (also the revocation cost), for the total number of members $N$ and the number of revoked members $R$. However, since a signature is required for each subset in the used subset difference method, the size is about $900R$ Bytes in the 128-bit security. In the case of $R = 100{,}000$, it amounts to about 80 MB. In this paper, we extend the scheme to reduce the revocation list (also the revocation cost). In the proposed scheme, an extended accumulator accumulates $T$ subsets, which is signed for the revocation list. The revocation list size is reduced by $1/T$, although the public key size, membership certificate size and the cost of a witness computation needed for signing increase related to $T$.

**Keywords:** Anonymity · Group signatures · Revocations · Accumulators

## 1 Introduction

The *group signature scheme* [13] allows a group member to anonymously sign a message on behalf of the group. In the group signature scheme, two types of trusted parties participate: A *group manager* ($GM$) has the authority to add a user to the own group. An *opener* can identify the signer from a signature. One of important issues in the group signature schemes is a *revocation* that the signing capability of a user is revoked. The revocation may happen, when the user leaves the group voluntarily or the account is banned due to the illegal usage, etc.

Lots of revocable group signature schemes have been proposed (e.g., [6–8,10–12,16,17,19,20]). Hereafter, let $N$ be the total number of group members, and $R$ be the number of revoked members. In the early scheme [7], the signature size is $O(R)$ (also, the costs of signing and verification). Then, the accumulator-based scheme has been proposed in [12], which is followed in [11], to achieve

This work was supported by JSPS KAKENHI Grant Number 25330153.


H.-S. Lee and D.-G. Han (Eds.): ICISC 2013, LNCS 8565, pp. 435–451, 2014.
DOI: 10.1007/978-3-319-12160-4_26

the constant-size signature with the constant verification costs. However, each member has to update a secret key (a witness for the accumulator) using the revocation data, which implies that signing costs is $O(R)$ in the worst case.

In [19], revocable schemes with the costs of constant signing and verification have been proposed. The demerit of the schemes is the long public key size. The basic scheme needs $O(N)$ size, and the extended one needs $O(\sqrt{N})$ in exchange for the extra signing cost. Recently, in [17], Libert et al. proposed an elegant scalable scheme using Naor et al.'s broadcast encryption framework [21]. This scheme achieves the constant verification cost, and the polylogarithmic public and secret key sizes. Finally, the same authors proposed the extended version with $O(1)$ secret key size [16], as achieving $O(1)$ signature size, $O(1)$ signing/verification costs and $O(\log N)$ public key size.

In this paper, we consider reducing the revocation list size. In [16], to indicate the revoked members, $O(R)$ size is needed for the revocation list. Furthermore, in the list, a signature is required for each subset in the used subset difference (SD) method, and the number of the signatures is bounded by $2R - 1$. The signature is an AHO signature [2], which needs 7 elements of a bilinear group. Assuming 128-bit security, the signature size is 448 Bytes. Thus, the revocation list size is about $900R$ Bytes or more. In an example of $R = 10{,}000$, the size amounts to 8 MB or more, and in case of $R = 100{,}000$, it becomes 80 MB or more. Note that the signer has to fetch all data of the latest revocation list every revocation epoch, as noted in [3]. This is because fetching a part of the list can reveal the information to trace the signer. Therefore, the large data may cause a delay in mobile environments.

In this paper, we propose a revocable group signature scheme with a compact revocation list as the extension of the state-of-the-art scheme [16]. In our scheme, using an extended accumulator based on [4], $GM$ accumulates $T$ subsets in the SD method, and signs the accumulated value. This is why the number of signatures is reduced by $1/T$. The revocation cost is similar. In case of $R = 100{,}000$, the size of the signature data including the accumulated value is reduced to 1,000 KB if $T = 100$. The compensation is increasing the public key size, the membership certificate size, and the cost of a witness computation needed for signing. Nevertheless, in case of $T = 100$, the public key size is 2,500 KB and the membership certificate size is 13 KB. In real applications, the public key and the certificate are not often distributed. On the other hand, the revocation list has to be distributed every revocation epoch. Thus, we consider that it is sufficiently practical to decrease the revocation list size while increasing the public key and the membership certificate sizes. The witness computation cost is about 120 exponentiations in case of $T = 100$. This cost is comparable to the computation cost of commitments in the original signing. This computation is needed only once every revocation epoch. As shown in Sect. 5, we can reduce the cost by computing only the modified parts from the previous epoch. Therefore, we consider that the extra costs are not a serious issue.

Due to the page limitation, the preliminary section reviewing the bilinear map and utilized primitives is in Appendix A.

## 2 Extended Accumulator

In [11], an efficient pairing-based accumulator is proposed. The accumulator is generated from a set of values, and we can verify that a single value is included in the set. In [22], the extended version is proposed, where we can verify that multiple values are included in the specified set, all at once. In [4], another extension is proposed, where we can verify that, for a set $U$, for all multiple sets $V_1, \ldots, V_T$, a value from $U$ is included in each $V_t$, i.e., $U \cap V_t \neq \emptyset$, all at once. This is applied to the verification for CNF formulas on attributes in the anonymous credential system of [4]. For a CNF formula $(\mathsf{a}_1 \in U \vee \cdots \vee \mathsf{a}_{L'} \in U) \wedge (\mathsf{b}_1 \in U \vee \cdots \vee \mathsf{b}_L \in U) \cdots$, setting $V_1 = \{\mathsf{a}_1, \ldots, \}, V_2 = \{\mathsf{b}_1, \ldots\}, \ldots$, we can verify the formula by checking $U \cap V_t \neq \emptyset$ for all $t$.

This paper furthermore extends the accumulator in [4], since our group signature scheme also needs the CNF-type verification. The scheme requires the verification of the logical formula as $(\mathsf{a}_{t1} \in U \wedge \cdots \wedge \mathsf{a}_{tL_t} \in U) \wedge (\mathsf{b}_{t1} \in U \vee \cdots \vee \mathsf{b}_{tL} \in U)$ for some $t$, given $V_t = \{\mathsf{a}_{t1}, \ldots, \mathsf{a}_{tL_t}\}, \tilde{V}_t = \{\mathsf{b}_{t1}, \ldots, \mathsf{b}_{tL}\}$ for all $1 \leq t \leq T$. The length of the AND relation is variable, but the length of the matched AND relation has to be hidden in the group signature scheme. Thus, we introduce a dummy parameter $\mathsf{SP}$. The other point of extension is to unbind the limitation of the number of given sets $(V_1, \tilde{V}_1), \ldots, (V_T, \tilde{V}_T)$, i.e., $2T$. In the previous accumulator, the number is bounded by the order $p$ of the bilinear groups. In our construction, for any $K, D$ s.t. $T = K \cdot D$, the target sets are divided to $((V_{1,1}, \tilde{V}_{1,1}), \ldots, (V_{1,D}, \tilde{V}_{1,D})), \ldots, ((V_{K,1}, \tilde{V}_{K,1}), \ldots, (V_{K,D}, \tilde{V}_{K,D}))$. Using randomized public parameters $(g_{k,1}, \ldots)$ for each $1 \leq k \leq K$, although $D$ is bounded by $p$, $T = K \cdot D$ becomes unbounded.

### 2.1 Proposed Construction

For all $1 \leq k \leq K$ and all $1 \leq d \leq D$, define $V_{k,d}$ and $\tilde{V}_{k,d}$ as subsets of $\{1, \ldots, n\}$. Define $\mathcal{V} = \{(V_{k,d}, \tilde{V}_{k,d})\}_{k=1,\ldots,K,d=1,\ldots,D}$. Let $U$ be a subset of $\{1, \ldots, n\}$ satisfying $U \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$ for some $1 \leq \tilde{d} \leq D$ and some $1 \leq \tilde{k} \leq K$. In this construction, we assume that the maximum of $|V_{k,d}|$ and $|\tilde{V}_{k,d}|$ is $\zeta$ for all $1 \leq k \leq K$ and all $1 \leq d \leq D$. In addition, we assume $(U \cap V_{k,d}) = (U \cap \tilde{V}_{k,d}) = \emptyset$ for all $1 \leq k \leq K$ and all $1 \leq d \leq D$ except some $k'$ and $d'$. If $U \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$, then it implies $k' = \tilde{k}$ and $d' = \tilde{d}$. These assumptions hold in our application to the revocable group signatures. We introduce mutually different special elements $\mathsf{SP}_{k,d} \in \mathcal{N}$ for all $k, d$ such that $\mathsf{SP}_{k,d} \notin V_{k',d'}$ for all $k', d'$. We assume that $\mathsf{SP}_{\tilde{k},\tilde{d}} \in U$ but $\mathsf{SP}_{k,d} \notin U$ for any $k \neq \tilde{k}, d \neq \tilde{d}$.

**AccSetup:** This is the algorithm to output the public parameters. The inputs are the security parameter $\mathsf{I}$ and $n, K, D, \{\mathsf{SP}_{k,d}\}_{1 \leq k \leq K, 1 \leq d \leq K}, \zeta$. Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order $p > 2^{\mathsf{I}}$ and a bilinear map $e$. Select

$g \in_R \mathcal{G}$. Select $\gamma, \eta_1, \ldots, \eta_K \in_R Z_p$, and compute $g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$, and $g_{k,1} = g_1^{\eta_k}, \ldots, g_{k,n} = g_n^{\eta_k}, g_{k,n+2} = g_{n+2}^{\eta_k}, \ldots, g_{k,2n} = g_{2n}^{\eta_k}$ $z_k = e(g,g)^{\eta_k \gamma^{n+1}}$ for all $1 \le k \le K$. For all $1 \le d \le D$, compute $c_d = (\zeta+1)^{2d-2}$, $\tilde{c}_d = (\zeta+1)^{2d-1}$ and set $\mathcal{C} = ((c_1, \tilde{c}_1), \ldots, (c_D, \tilde{c}_D))$. We assume that $(\zeta+1)c_D < p$. Publish $n, K, D, \{\mathsf{SP}_{k,d}\}_{1 \le k \le K, 1 \le d \le K}, \zeta, \mathcal{C}, p, \mathcal{G}, \mathcal{T}, e, g, (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}), \{g_{k,1}, \ldots, g_{k,n}, g_{k,n+2}, \ldots, g_{k,2n}, z_k\}_{k=1}^K$ as the public parameters.

**AccGen:** This is the algorithm to compute the accumulator using the public parameters. The accumulator $acc_{\mathcal{V}}$ of $\mathcal{V}$ is computed as

$$acc_{\mathcal{V}} = \prod_{1 \le k \le K} \prod_{1 \le d \le D} ((\prod_{j \in V_{k,d}} g_{k,n+1-j})^{c_d} \cdot (\prod_{j=1}^{\zeta - |V_{k,d}|} g_{k,n+1-\mathsf{SP}_{k,d}})^{c_d} \cdot (\prod_{j \in \tilde{V}_{k,d}} g_{k,n+1-j})^{\tilde{c}_d}).$$

**AccWitGen:** This is the algorithm to compute the witness that $U \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$ for some $1 \le \tilde{d} \le D$ and some $1 \le \tilde{k} \le K$, using the public parameters. Given $U$, $\mathcal{V}$, and the accumulator $acc_{\mathcal{V}}$, the witness is computed as

$$W = \prod_{i \in U} \prod_{1 \le k \le K} \prod_{1 \le d \le D} ((\prod_{j \in V_{k,d}}^{j \neq i} g_{k,n+1-j+i})^{c_d} \cdot (\prod_{j=1}^{\zeta - |V_{k,d}|, i \neq \mathsf{SP}_{k,d}} g_{k,n+1-\mathsf{SP}_{k,d}+i})^{c_d} \cdot (\prod_{j \in \tilde{V}_{k,d}}^{j \neq i} g_{k,n+1-j+i})^{\tilde{c}_d}).$$

Furthermore, the auxiliary parameters are set as $\tilde{k}, \tilde{d}$, $\delta_{\tilde{k},\tilde{d}} = |U \cap \tilde{V}_{\tilde{k},\tilde{d}}|$.

**AccVerify:** This is the algorithm to verify that $U \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$ for some $1 \le \tilde{d} \le D$ and some $1 \le \tilde{k} \le K$, using the witness, the auxiliary parameters, and the public parameters. Given $acc_{\mathcal{V}}, U$, $W$ $\tilde{k}, \tilde{d}$ and $\delta_{\tilde{k},\tilde{d}}$, accept if

$$\frac{e(\prod_{i \in U} g_i, acc_{\mathcal{V}})}{e(g, W)} = z_{\tilde{k}}^{\zeta c_{\tilde{d}} + \delta_{\tilde{k},\tilde{d}} \tilde{c}_{\tilde{d}}}, \qquad 1 \le \delta_{\tilde{k},\tilde{d}} \le \zeta. \tag{1}$$

### 2.2 Security

We can show the correctness and the security. The proofs are shown in the full paper.

**Theorem 1.** *Assume that* **AccSetup**, **AccGen**, **AccWitGen** *correctly compute all parameters. Then,* **AccVerify** *accepts* $U, acc_{\mathcal{V}}, W, \tilde{k}, \tilde{d}$ *and* $\delta_{\tilde{k},\tilde{d}}$ *that they outputs.*

**Theorem 2.** *Under the n-DHE assumption, any adversary cannot output* $(U, \mathcal{V}, W, \tilde{k}, \tilde{d}, \delta_{\tilde{k},\tilde{d}})$, *on inputs* $n, K, D, \{\mathsf{SP}_{k,d}\}_{1 \le k \le K, 1 \le d \le K}, \zeta, \mathcal{C}, p, \mathcal{G}, \mathcal{T}, e, g, (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}), \{g_{k,1}, \ldots, g_{k,n}, g_{k,n+2}, \ldots, g_{k,2n}, z_k\}_{k=1}^K$ *s.t.* **AccVerify** *accepts* $U, acc_{\mathcal{V}}, W, \tilde{k}, \tilde{d}, \delta_{\tilde{k},\tilde{d}}$ *but* $U \cap V_{k',d'} \neq V_{k',d'}$ *or* $U \cap \tilde{V}_{k',d'} = \emptyset$ *for some* $k', d'$, *assuming the following preconditions.*

1. $(U \cap V_{k,d}) = (U \cap \tilde{V}_{k,d}) = \emptyset$ *for all* $1 \leq k \leq K$ *and all* $1 \leq d \leq D$ *except* $k = k'$ *and* $d = d'$,
2. *only* $\mathsf{SP}_{k',d'}$ *is included in* $U$ *(other* $\mathsf{SP}_{k,d}$ *is not included).*

## 3 Syntax and Security of Revocable Group Signatures

### 3.1 Syntax

**Setup**$(\mathsf{l}, N, K, D)$**:** Given the security parameter $\mathsf{l} \in \mathbb{N}$, the maximum number of group members $N \in \mathbb{N}$, and the efficiency parameters $K, D \in \mathbb{N}$, this algorithm outputs a group public key $gpk$, a $GM$'s secret key $gsk$, and an opener's secret key $osk$. This algorithm initializes a public state $St$ comprising a set data structure $St_{users} = \emptyset$ and a string data structure $St_{trans} = \epsilon$.

**Join:** This is an interactive protocol between the group manager $GM$ and a joining user $\mathcal{U}_{\mathsf{i}}$. The interactive Turing machines are denoted as $\mathsf{J}_{GM}$ and $\mathsf{J}_{\mathcal{U}_{\mathsf{i}}}$, respectively. After the protocol $[\mathsf{J}_{GM}(\mathsf{l}, St, gpk, gsk), \mathsf{J}_{\mathcal{U}_{\mathsf{i}}}(\mathsf{l}, gpk)]$ is executed, $\mathsf{J}_{\mathcal{U}_{\mathsf{i}}}$ outputs a membership secret $sec_{\mathsf{i}}$ and a membership certificate $cert_{\mathsf{i}}$. The protocol is successful, $\mathsf{J}_{GM}$ updates $St$ by setting $St_{user} = St_{user} \cup \{\mathsf{i}\}$ and $St_{trans} = St_{trans} \| \langle \mathsf{i}, \mathsf{transcript}_{\mathsf{i}} \rangle$.

**Revoke**$(gpk, gsk, \tau, \mathcal{R}_\tau)$**:** Given $gpk, gsk$, epoch $\tau$ and $\mathcal{R}_\tau \subset \{1, \ldots, N\}$ that is the identities of revoked members at the epoch $\tau$, this algorithm outputs the revocation list $RL_\tau$.

**Sign**$(gpk, \tau, RL_\tau, cert_{\mathsf{i}}, sec_{\mathsf{i}}, M)$**:** Given $gpk, \tau, RL_\tau$, the signing member's $cert_{\mathsf{i}}$, $sec_{\mathsf{i}}$, and the message $M$ to be signed, this algorithm outputs $\perp$ if $i \in \mathcal{R}_t$ or the signature $\sigma$ otherwise.

**Verify**$(gpk, \tau, RL_\tau, \sigma, M)$**:** Given $gpk, \tau, RL_\tau$, the signature $\sigma$ and message $M$, this algorithm outputs 1 if the signature is valid and not revoked for the revocation list $RL_\tau$, or 0 otherwise.

**Open**$(gpk, \tau, RL_\tau, \sigma, M, St, osk)$**:** Given $gpk, \tau, RL_\tau, \sigma, M$ as in **Verify**, the state $St$ in **Join**, and the opener's secret key $osk$, this algorithm outputs $\mathsf{i} \in St_{users} \cup \{\perp\}$ which means the identity of the signer of $\sigma$ or a symbol of an opening failure.

### 3.2 Security Model

The security of the revocable group signature scheme consists of *security against misidentification attacks*, *security against framing attacks*, and *anonymity*. The security against misidentification attacks requires that the adversary cannot forge a signature that is identified to one outside the set of corrupted and non-revoked members. The security against framing attacks requires that a signature of an honest member cannot be computed by other members and even $GM$. The anonymity captures the anonymity and the unlinkability of signatures. The formal definitions are described in the full paper.

# 4 A Revocable Group Signature with Compact Revocation List and Constant Verification Time

## 4.1 Construction Idea

The proposed scheme is based on the previous scheme [16]. The approach of the previous scheme is as follows. The subset cover framework with the SD method is used. To each member, a leaf node $v$ in the binary tree with the height $L$ for $N = 2^L$ is assigned. Every node in the tree is assigned to a unique number. In **Join**, to the member, a membership certificate is issued, which is an AHO signature on a public key and an accumulated data for the node numbers on the path from the root to $v$, $\mathsf{ID}_1, \ldots, \mathsf{ID}_L$. For the accumulation, they adopt a vector commitment [18] that is similar to the accumulators. In **Revoke**, $GM$ publishes the revocation list, where each entry consists of accumulated values for primary and secondary nodes in each $S_i$ in the SD method, and the AHO signature on them and the current time epoch $\tau$. In the group signature, to show that the signer is not a revoked member, she proves

1. an AHO signature binds between $\tau$ and the primary node with number $\tilde{\mathsf{ID}}_{i,\phi_i}$ of level $\phi_i$ and the secondary node $\tilde{\mathsf{ID}}_{i,\psi_i}$ of level $\psi_i$ in an $S_i$,
2. for $\mathsf{ID}_{\phi_i}$ with level $\phi_i$ and $\mathsf{ID}_{\psi_i}$ with level $\psi_i$ in the membership certificate, it holds that $\mathsf{ID}_{\phi_i} = \tilde{\mathsf{ID}}_{i,\phi_i}$ and $\mathsf{ID}_{\psi_i} \neq \tilde{\mathsf{ID}}_{i,\psi_i}$.

The second relation means that the primary node $\tilde{\mathsf{ID}}_{i,\phi_i}$ is an ancestor of $v$ and the secondary node $\tilde{\mathsf{ID}}_{i,\psi_i}$ is not, i.e., the subset $S_i$ includes $v$, which implies that the member is not revoked due to the subset cover framework. In this approach, an AHO signature is needed for each subset $S_i$. Each signature needs long data (448 Bytes in 128-bit security), and thus the revocation list becomes long as $R$ increases.

In our approach, to accumulate the revocation list, we adopt the extended accumulator in Sect. 2. Although the same tree structure in the subset cover framework is used, a different coding is used. In the tree, for the edge to the left (resp., right) child in the depth $j$, use index $(j, 0)$ (resp, $(j, 1)$). Then, for the leaf $v$ assigned to the member, let $(1, x_1), \ldots, (L, x_L)$ be the path from the root to the leaf $v$, where $x_\ell \in \{0, 1\}$. Similarly, for the subset $S_i$, let $(1, s_{i,1}), \ldots, (\phi_i, s_{i,\phi_i})$ denote the path from the root to the primary root and let $(1, s_{i,1}), \ldots, (\psi_i, s_{i,\psi_i})$ denote the path to the secondary root, where $\phi_i, \psi_i \in \{1, \ldots, L\}$ and $s_{i,j} \in \{0, 1\}$. To prove the non-revocation, the signer prove that $((1, x_1) = (1, s_{i_1})) \wedge \cdots \wedge ((\phi_i, x_{\phi_i}) = (\phi_i, s_{\phi_i}))$ (i.e., the primary node is an ancestor $v$) and $((\phi_i + 1, x_{\phi_i+1}) \neq (\phi_i + 1, s_{\phi_i+1})) \vee \cdots \vee ((\psi_i, x_{\psi_i}) \neq (\psi_i, s_{\psi_i}))$ (i.e., the secondary node is not an ancestor of $v$). The latter relation can be rewritten as $((\phi_i + 1, x_{\phi_i+1}) = (\phi_i + 1, \overline{s_{\phi_i+1}})) \vee \cdots \vee ((\psi_i, x_{\psi_i}) = (\psi_i, \overline{s_{\psi_i}}))$.

Using the accumulator, we can prove the relations. Let $T$ be the number of accumulated $S_i$. For $T$, given $K, D$ such that $T = K \cdot D$. For all $1 \leq t \leq T$, consider function $I_t$ mapping $\{(\ell, b)\}_{1 \leq \ell \leq L, b \in \{0,1\}}$ to $\{T + 1, \ldots, n\}$ such that $\{I_t(\ell, b)\}_{1 \leq \ell \leq L, b \in \{0,1\}} \cap \{I_{t'}(\ell, b)\}_{1 \leq \ell \leq L, b \in \{0,1\}} = \emptyset$ for any pair $1 \leq t, t' \leq T$.

Set $\mathsf{SP}_{k,d} = D \cdot (k-1) + d$ for all $1 \leq k \leq K$ and $1 \leq d \leq D$. Note that $\mathsf{SP}_{k,d} \in \{1, \ldots, T\}$. The relation is required to satisfy the precondition of the accumulator. Define $U_t = \{I_t(1, x_1), \ldots, I_t(L, x_L), \mathsf{SP}_{k,d}\}$ for all $1 \leq t \leq T$, where $k = \lceil t/D \rceil$ and $d = t \bmod D$. The accumulated $P_t = \prod_{i \in U_t} g_i$ is embedded into a membership certificate for all $t$. As for the revocation list, for $w = \lceil m/T \rceil$, divide $S_1, \ldots, S_m$ into $w$ sequences:

$$\mathcal{S}_1 = (S_1, \ldots, S_T), \mathcal{S}_2 = (S_{T+1}, \ldots, S_{2T}), \ldots, \mathcal{S}_w = (S_{(w-1)T+1}, \ldots, S_m),$$

where $\mathcal{S}_1, \ldots, \mathcal{S}_{w-1}$ contain $T$ elements and $\mathcal{S}_w$ contains $T$ or less elements. Here, we can connect any $S_i$ to the corresponding sequence $\mathcal{S}_\omega$ by the relation $\omega = \lceil i/T \rceil$. For each $\mathcal{S}_\omega$, do the following. Compute $t = i \bmod T$ to determine the position of $S_i$ in $\mathcal{S}_\omega$. Transform $t$ to the corresponding $(k, d)$ in the accumulator, by $k = \lceil t/D \rceil$ and $d = t \bmod D$. For all $(k, d)$ correspondent $1 \leq t \leq T$ in $\mathcal{S}_\omega$ (i.e., $(\omega - 1)T + 1 \leq i \leq \omega T$), set $V_{k,d} = \{I_t(1, s_{i,1}), \ldots, I_t(\phi_i, s_{i,\phi_i})\}$ and $\tilde{V}_{k,d} = \{I_t(\phi_i + 1, \overline{s_{i,\phi_i+1}}), \ldots, I_t(\psi_i, \overline{s_{i,\psi_i}})\}$. As the revocation list, $GM$ publishes the accumulator $acc_{\mathcal{V}}$ for $\mathcal{V} = \{(V_{k,d}, \tilde{V}_{k,d})\}_{k=1,\ldots,K,d=1,\ldots,D}$ together with the AHO signature. By accumulating $S_i$'s into $\mathcal{S}_\omega$, the number of published signatures is reduced by $1/T$.

In the group signature, for some $\tilde{t}$, the signer proves that $U_{\tilde{t}} \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U_{\tilde{t}} \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$ for some $1 \leq \tilde{d} \leq D$ and some $1 \leq \tilde{k} \leq K$, using the accumulator verification. The former relation means the AND relation $((1, x_1) = (1, s_{i_1})) \wedge \cdots$ and the latter means that OR relation $((\phi_i + 1, x_{\phi_i+1}) = (\phi_i + 1, \overline{s_{\phi_i+1}})) \vee \cdots$. In the verification relations (1) of the accumulator, the right hand reveals the indexes $\tilde{k}, \tilde{d}$ via $z_{\tilde{k}}, c_{\tilde{d}}, \tilde{c}_{\tilde{d}}$. To hide the indexes, we utilize the technique of membership proof using signatures [9]. Also, we utilize the technique to prove $1 \leq \delta_{\tilde{k},\tilde{d}} \leq \zeta$ in the accumulator.

### 4.2 Proposed Construction

**Setup.** The inputs are the security parameter $\mathsf{l}$, the maximum number of group members $N$, and the efficiency parameters $K, D$.

1. Select bilinear groups $\mathcal{G}$, $\mathcal{T}$ with the same order $p > 2^{\mathsf{l}}$ and the bilinear map $e$, and $g \in_R \mathcal{G}$.
2. Set parameter $T = K \cdot D$.
3. Generate public parameters of the extended accumulator: Set $\zeta = L$. Set $\mathsf{SP}_{k,d} = D \cdot (k-1) + d$ for all $1 \leq k \leq K$ and $1 \leq d \leq D$. Note that $\mathsf{SP}_{k,d} \in \{1, \ldots, T\}$. Select $\gamma, \eta_1, \ldots, \eta_K \in_R Z_p$, and compute $g_1 = g^{\gamma^1}, \ldots, g_n = g^{\gamma^n}, g_{n+2} = g^{\gamma^{n+2}}, \ldots, g_{2n} = g^{\gamma^{2n}}$, and $g_{k,1} = g_1^{\eta_k}, \ldots, g_{k,n} = g_n^{\eta_k}, g_{k,n+2} = g_{n+2}^{\eta_k}, \ldots, g_{k,2n} = g_{2n}^{\eta_k}$ $z_k = e(g, g)^{\eta_k \gamma^{n+1}}$ for all $1 \leq k \leq K$. For all $1 \leq d \leq D$, compute $c_d = (\zeta + 1)^{2d-2}$, $\tilde{c}_d = (\zeta + 1)^{2d-1}$ and set $\mathcal{C} = ((c_1, \tilde{c}_1), \ldots, (c_D, \tilde{c}_D))$. Set

$$pk_{\mathrm{acc}} = (\{\mathsf{SP}_{k,d}\}_{1 \leq k \leq K, 1 \leq d \leq D}, \zeta, \mathcal{C}, (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}), \{g_{k,1}, \ldots, g_{k,n}, g_{k,n+2}, \ldots, g_{k,2n}, z_k\}_{k=1}^{K}).$$

4. Define $n_1 = n_3 = n_4 = 2, n_2 = 1$. Generate four key pairs for the AHO signature:

$$pk_{\mathrm{AHO}}^{(d)} = (G_r^{(d)}, H_r^{(d)}, G_z^{(d)}, H_z^{(d)}, \{G_i^{(d)}, H_i^{(d)}\}_{i=1}^{n_d}, A^{(d)}, B^{(d)}),$$
$$sk_{\mathrm{AHO}}^{(d)} = (\alpha_a^{(d)}, \alpha_b^{(d)}, \mu_z^{(d)}, \nu_z^{(d)}, \mu, \nu),$$

   where $d \in \{1, 2, 3, 4\}$.
5. Generate a CRS for the GS NIWI proof: select $\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$, where $\boldsymbol{f}_1 = (f_1, 1, g)$, $\boldsymbol{f}_2 = (1, f_2, g)$, $\boldsymbol{f}_3 = \boldsymbol{f}_1^{\xi_1} \cdot \boldsymbol{f}_2^{\xi_2}$ for $\xi_1, \xi_2, y_1, y_2 \in_R Z_p^*$ and $f_1 = g^{y_1}, f_2 = g^{y_2}$. Set $\tilde{\boldsymbol{f}} = \boldsymbol{f}_3 \cdot (1, 1, g)$.
6. Define set $\Phi = \{(g_{k,1}^{c_d}, g_{k,1}^{\tilde{c}_d}) | 1 \le k \le K, 1 \le d \le D\}$, where $|\Phi| = K \cdot D = T$. For every $(g_{k,1}^{c_d}, g_{k,1}^{\tilde{c}_d}) \in \Phi$, generate the AHO signature on two messages $(g_{k,1}^{c_d}, g_{k,1}^{\tilde{c}_d})$, using $sk_{\mathrm{AHO}}^{(1)}$. The signature is denoted as $\tilde{\sigma}_t = (\tilde{\theta}_{t1}, \ldots, \tilde{\theta}_{t7})$, where $t = D \cdot (k-1) + d$.
7. For every $1 \le \delta \le \zeta$, generate the AHO signature on message $g_n^\delta$, using $sk_{\mathrm{AHO}}^{(2)}$. The signature is denoted as $\hat{\sigma}_\delta = (\hat{\theta}_{\delta 1}, \ldots, \hat{\theta}_{\delta 7})$.
8. Select $\mathcal{U}, \mathcal{V} \in_R \mathcal{G}$ for a pubic encryption.
9. Select a strongly unforgeable one-time signature $\Sigma_{\mathrm{OTS}} = (\mathbf{Setup}_{\mathrm{OTS}}, \mathbf{Sign}_{\mathrm{OTS}}, \mathbf{Verify}_{\mathrm{OTS}})$.
10. Output the group public key $gpk = (K, D, p, \mathcal{G}, \mathcal{T}, e, g, pk_{\mathrm{acc}}, \{pk_{\mathrm{AHO}}^{(i)}\}_{i=1,2,3,4}, \boldsymbol{f}, \tilde{\boldsymbol{f}}, \{\tilde{\sigma}_t\}_{t \in \Phi}, \{\hat{\sigma}_\delta\}_{1 \le \delta \le \zeta}, (\mathcal{U}, \mathcal{V}), \Sigma_{\mathrm{OTS}})$, the $GM$'s secret key $gsk = (\{sk_{\mathrm{AHO}}^{(i)}\}_{i=1,2,3,4})$ and the opener's secret key $osk = (y_1, y_2)$.

**Join.** The common inputs of $\mathsf{J}_{GM}$ and $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ are $\mathsf{I}, gpk$. The additional inputs of $\mathsf{J}_{GM}$ are $St, gsk$.

1. $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ selects $x \in_R \mathcal{G}$, computes $X = g^x$ and send $X$ to $\mathsf{J}_{GM}$. If $X$ is already registered in database $St_{trans}$, $\mathsf{J}_{GM}$ halts and returns $\perp$ to $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$.
2. $\mathsf{J}_{GM}$ assigns to the user a leaf $v$ in the tree. Let $(1, x_1), \ldots, (L, x_L)$ be the path from the root to the leaf $v$. Define $U_t = \{I_t(1, x_1), \ldots, I_t(L, x_L), \mathsf{SP}_{k,d}\}$ for all $1 \le t \le T$, where $k = \lceil t/D \rceil$ and $d = t \bmod D$. $\mathsf{J}_{GM}$ computes $P_t = \prod_{i \in U_t} g_i$ for all $1 \le t \le T$.
3. $\mathsf{J}_{GM}$ generates an AHO signature $\sigma_t = (\theta_{t,1}, \ldots, \theta_{t,7})$ on $(X, P_t)$ for all $1 \le t \le T$, using $sk_{\mathrm{AHO}}^{(3)}$.
4. $\mathsf{J}_{GM}$ sends $v, \{P_t\}_{1 \le t \le T}$ to $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$. $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ checks the correctness of $P_t$'s. If these are incorrect, $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ aborts. Otherwise, $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ sends $\mathsf{J}_{GM}$ the ordinary digital signature $sig_\mathsf{i}$ on $(X, v)$.
5. $\mathsf{J}_{GM}$ verifies $sig$. If it is incorrect, $\mathsf{J}_{GM}$ aborts. Otherwise, $\mathsf{J}_{GM}$ sends the AHO signature $\sigma_t$ to $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$, and stores $\langle \mathsf{i}, \mathsf{transcript}_\mathsf{i} = (v, X, \{P_t, \sigma_t\}_{1 \le t \le T}, sig_\mathsf{i}) \rangle$ in the database $St_{trans}$.
6. $\mathsf{J}_{\mathcal{U}_\mathsf{i}}$ outputs the membership certificate $cert_\mathsf{i} = (v, X, \{U_t, P_t, \sigma_t\}_{1 \le t \le T})$ and the membership secret $sec_\mathsf{i} = x$.

**Revoke.** The inputs are $gpk, gsk$, the epoch $\tau$ and the revocation members $\mathcal{R}_\tau$.

1. By the subset covering of the SD scheme, find a cover of the unrevoked users, $S_1, \ldots, S_m$. Set $w = \lceil m/T \rceil$. Divide $S_1, \ldots, S_m$ into $w$ sequences:

$$\mathcal{S}_1 = (S_1, \ldots, S_T), \mathcal{S}_2 = (S_{T+1}, \ldots, S_{2T}), \ldots, \mathcal{S}_w = (S_{(w-1)T+1}, \ldots, S_m),$$

   where $\mathcal{S}_1, \ldots, \mathcal{S}_{w-1}$ contain $T$ elements and $\mathcal{S}_w$ contains $T$ or less elements. Here, we can connect any $S_i$ to the corresponding sequence $\mathcal{S}_\omega$ by the relation $\omega = \lceil i/T \rceil$. For the sub-tree $S_i$, let $(1, s_{i,1}), \ldots, (\phi_i, s_{i,\phi_i})$ denote the path from the root to the primary root and let $(1, s_{i,1}), \ldots, (\psi_i, s_{i,\psi_i})$ denote the path to the secondary root, where $\phi_i, \psi_i \in \{1, \ldots, L\}$ and each $s_{i,j} \in \{0, 1\}$.
2. For $\mathcal{S}_\omega$ with all $1 \leq \omega \leq w$, do the following.
   (a) To determine the position of $S_i$ in $\mathcal{S}_\omega$, compute $t = i \bmod T$. Transform $t$ to the corresponding $(k, d)$ in the accumulator, by $k = \lceil t/D \rceil$ and $d = t \bmod D$. For all $(k, d)$ correspondent $1 \leq t \leq T$ in $\mathcal{S}_\omega$ (i.e., $(\omega - 1)T + 1 \leq i \leq \omega T$), set $V_{k,d} = \{I_t(1, s_{i,1}), \ldots, I_t(\phi_i, s_{i,\phi_i})\}$ and $\tilde{V}_{k,d} = \{I_t(\phi_i + 1, \overline{s_{i,\phi_i+1}}), \ldots, I_t(\psi_i, \overline{s_{i,\psi_i}})\}$, where $\overline{s_{i,\ell}}$ is the negation of $s_{i,\ell}$.
   (b) Compute $acc_\omega = \prod_{1 \leq k \leq K} \prod_{1 \leq d \leq D} ((\prod_{j \in V_{k,d}} g_{k,n+1-j})^{c_d} \cdot (\prod_{j=1}^{\zeta - |V_{k,d}|} g_{k,n+1-\mathsf{SP}_{k,d}})^{c_d} \cdot (\prod_{j \in \tilde{V}_{k,d}} g_{k,n+1-j})^{\tilde{c}_d})$.
3. For all $1 \leq \omega \leq w$, compute the AHO signature on pair $(g^\tau, acc_\omega)$: $\Theta_\omega = (\Theta_{\omega,1}, \ldots, \Theta_{\omega,7})$, using $sk^{(4)}_{\mathrm{AHO}}$.
4. Output the revocation list: $RL_\tau = (\tau, \mathcal{R}_\tau, \{S_i\}_{i=1}^m, \{acc_\omega, \Theta_\omega\}_{\omega=1}^w)$.

**Sign.** The inputs are $gpk, \tau, RL_\tau, cert_{\mathrm{i}}, sec_{\mathrm{i}}$ and the message $M$.

1. Using $\mathbf{Setup}_{\mathrm{OTS}}$, generate a key pair $(\mathsf{SK}, \mathsf{VK})$ of the one-time signature.
2. Using $RL_\tau$, find the set $S_{\tilde{\imath}}$ including the signing user. For the subset $S_{\tilde{\imath}}$, let $(1, s_{\tilde{\imath},1}), \ldots, (\phi_{\tilde{\imath}}, s_{\tilde{\imath},\phi_{\tilde{\imath}}})$ denote the path from the root to the primary root and let $(1, s_{\tilde{\imath},1}), \ldots, (\psi_{\tilde{\imath}}, s_{\tilde{\imath},\psi_{\tilde{\imath}}})$ denote the path to the secondary root. Then, find $\mathcal{S}_{\tilde{\omega}}$ including $S_{\tilde{\imath}}$ by $\tilde{\omega} = \lceil \tilde{\imath}/T \rceil$. To determine the position of $S_{\tilde{\imath}}$ in $\mathcal{S}_{\tilde{\omega}}$, compute $\tilde{t} = \tilde{\imath} \bmod T$. Furthermore, find the corresponding $(\tilde{k}, \tilde{d})$ by $\tilde{k} = \lceil \tilde{t}/D \rceil$ and $\tilde{d} = \tilde{t} \bmod D$ satisfying $\tilde{t} = D \cdot (\tilde{k} - 1) + \tilde{d} - 1$.
3. Pick up $acc_{\tilde{\omega}}, \Theta_{\tilde{\omega}} = (\Theta_{\tilde{\omega},1}, \ldots, \Theta_{\tilde{\omega},7})$ from $RL_\tau$, and $U_{\tilde{t}}, P_{\tilde{t}}, \sigma_{\tilde{t}} = (\theta_{\tilde{t},1}, \ldots, \theta_{\tilde{t},7})$ from $cert_{\mathrm{i}}$. For $\tilde{t}, \tilde{k}, \tilde{d}$, pick up the AHO signature on $(J_{\tilde{t}1}, J_{\tilde{t}2}) = (g_{\tilde{k},1}^{c_{\tilde{d}}}, g_{\tilde{k},1}^{\tilde{c}_{\tilde{d}}})$, i.e., $\tilde{\sigma}_{\tilde{t}} = (\tilde{\theta}_{\tilde{t}1}, \ldots, \tilde{\theta}_{\tilde{t}7})$ from $gpk$. In the same way to **Revoke**, set $V_{k,d}$ and $\tilde{V}_{k,d}$ for all $(k, d)$ in $\mathcal{S}_{\tilde{\omega}}$. Compute $\delta_{\tilde{k},\tilde{d}} = |U_{\tilde{t}} \cap \tilde{V}_{\tilde{k},\tilde{d}}|$. Pick up the AHO signature on $Q_{\delta_{\tilde{k},\tilde{d}}} = g_n^{\delta_{\tilde{k},\tilde{d}}}$, i.e., $\hat{\sigma}_{\delta_{\tilde{k},\tilde{d}}} = (\hat{\theta}_{\delta_{\tilde{k},\tilde{d}}1}, \ldots, \hat{\theta}_{\delta_{\tilde{k},\tilde{d}}7})$ from $gpk$.
4. Compute the witness of $U_{\tilde{t}} \cap V_{\tilde{k},\tilde{d}} = V_{\tilde{k},\tilde{d}}$ and $U_{\tilde{t}} \cap \tilde{V}_{\tilde{k},\tilde{d}} \neq \emptyset$, as follows. $W = \prod_{i \in U} \prod_{1 \leq k \leq K} \prod_{1 \leq d \leq D} ((\prod_{j \in V_{k,d}}^{j \neq i} g_{k,n+1-j+i})^{c_d} \cdot (\prod_{j=1}^{\zeta - |V_{k,d}|, i \neq \mathsf{SP}_{k,d}} g_{k,n+1-\mathsf{SP}_{k,d}+i})^{c_d} \cdot (\prod_{j \in \tilde{V}_{k,d}}^{j \neq i} g_{k,n+1-j+i})^{\tilde{c}_d})$.
5. Compute GS commitments $com_{P_{\tilde{t}}}, com_{acc_{\tilde{\omega}}}, com_W, com_{J_{\tilde{t}1}}, com_{J_{\tilde{t}2}}, com_{Q_{\delta_{\tilde{k},\tilde{d}}}}$, $com_X$ to $P_{\tilde{t}}$, $acc_{\tilde{\omega}}, W$, $J_{\tilde{t}1}, J_{\tilde{t}2}$, $Q_{\delta_{\tilde{k},\tilde{d}}}, X$. Then, re-randomize the AHO signatures $\sigma_{\tilde{t}}, \tilde{\sigma}_{\tilde{t}}, \hat{\sigma}_{\delta_{\tilde{k},\tilde{d}}}, \Theta_{\tilde{\omega}}$ to obtain $\sigma'_{\tilde{t}} = \{\theta'_1, \ldots, \theta'_7\}$, $\tilde{\sigma'}_{\tilde{t}} = \{\tilde{\theta'}_1, \ldots, \tilde{\theta'}_7\}$, $\hat{\sigma'}_{\delta_{\tilde{k},\tilde{d}}} =$

$\{\hat{\theta'}_1, \ldots, \hat{\theta'}_7\}, \Theta'_{\tilde{\omega}} = \{\Theta'_1, \ldots, \Theta'_7\}$, and compute GS commitments $\{com_{\theta'_i}\}_{i\in\{1,2,5\}}$, $\{com_{\tilde{\theta'}_i}\}_{i\in\{1,2,5\}}$, $\{com_{\hat{\theta'}_i}\}_{i\in\{1,2,5\}}$, $\{com_{\Theta'_i}\}_{i\in\{1,2,5\}}$ to $\{\theta'_i\}_{i\in\{1,2,5\}}$, $\{\tilde{\theta'}i\}_{i\in\{1,2,5\}}$, $\{\hat{\theta'}i\}_{i\in\{1,2,5\}}, \{\Theta'_i\}_{i\in\{1,2,5\}}$.

6. Generate $\{\pi_i\}_{i=1}^{9}$ s.t.

$$1_{\mathcal{T}} = e(P_{\tilde{t}}, acc_{\tilde{\omega}}) \cdot e(g, W)^{-1} \cdot e(J_{\tilde{t}1}, g_n^{\zeta})^{-1} \cdot e(J_{\tilde{t}2}, Q_{\delta_{\tilde{k},\tilde{d}}})^{-1}, \tag{2}$$

$$A^{(1)} \cdot e(\tilde{\theta}'_3, \tilde{\theta}'_4)^{-1} = e(G_z^{(1)}, \tilde{\theta}'_1) \cdot e(G_r^{(1)}, \tilde{\theta}'_2) \cdot e(G_1^{(1)}, J_{\tilde{t}1}) \cdot e(G_2^{(1)}, J_{\tilde{t}2}), \tag{3}$$

$$B^{(1)} \cdot e(\tilde{\theta}'_6, \tilde{\theta}'_7)^{-1} = e(H_z^{(1)}, \tilde{\theta}'_1) \cdot e(H_r^{(1)}, \tilde{\theta}'_5) \cdot e(H_1^{(1)}, J_{\tilde{t}1}) \cdot e(H_2^{(1)}, J_{\tilde{t}2}), \tag{4}$$

$$A^{(2)} \cdot e(\hat{\theta}'_3, \hat{\theta}'_4)^{-1} = e(G_z^{(2)}, \hat{\theta}'_1) \cdot e(G_r^{(2)}, \hat{\theta}'_2) \cdot e(G_1^{(2)}, Q_{\delta_{\tilde{k},\tilde{d}}}), \tag{5}$$

$$B^{(2)} \cdot e(\hat{\theta}'_6, \hat{\theta}'_7)^{-1} = e(H_z^{(2)}, \hat{\theta}'_1) \cdot e(H_r^{(2)}, \hat{\theta}'_5) \cdot e(H_1^{(2)}, Q_{\delta_{\tilde{k},\tilde{d}}}), \tag{6}$$

$$A^{(3)} \cdot e(\theta'_3, \theta'_4)^{-1} = e(G_z^{(3)}, \theta'_1) \cdot e(G_r^{(3)}, \theta'_2) \cdot e(G_1^{(3)}, X) \cdot e(G_2^{(3)}, P_{\tilde{t}}), \tag{7}$$

$$B^{(3)} \cdot e(\theta'_6, \theta'_7)^{-1} = e(H_z^{(3)}, \theta'_1) \cdot e(H_r^{(3)}, \theta'_5) \cdot e(H_1^{(3)}, X) \cdot e(H_2^{(3)}, P_{\tilde{t}}), \tag{8}$$

$$A^{(4)} \cdot e(\Theta'_3, \Theta'_4)^{-1} \cdot e(G_1^{(4)}, g^{\tau})^{-1} = e(G_z^{(4)}, \Theta'_1) \cdot e(G_r^{(4)}, \Theta'_2) \cdot e(G_2^{(4)}, acc_{\tilde{\omega}}), \tag{9}$$

$$B^{(4)} \cdot e(\Theta'_6, \Theta'_7)^{-1} \cdot e(H_1^{(4)}, g^{\tau})^{-1} = e(H_z^{(4)}, \Theta'_1) \cdot e(H_r^{(4)}, \Theta'_5) \cdot e(H_2^{(4)}, acc_{\tilde{\omega}}). \tag{10}$$

In the GS proofs, the Eq. (2) shows the accumulator verification, the Eqs. (3), (4) shows the AHO signature verification on $(J_{\tilde{t}1}, J_{\tilde{t}2})$, the Eqs. (5), (6) shows the AHO signature verification on $Q_{\delta_{\tilde{k},\tilde{d}}}$, the Eqs. (7), (8) shows the AHO signature verification on $(X, P_{\tilde{t}})$, and the Eqs. (9), (10) shows the AHO signature verification on $(g^{\tau}, acc_{\tilde{\omega}})$.

7. The remaining process is as the same as in [16]. Using $\mathsf{VK}$ as a tag, compute a tag-based encryption [15] of $X$. Namely, select $z_1, z_2 \in Z_p$, and compute

$$(\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot \mathcal{U})^{z_1}, (g^{\mathsf{VK}} \cdot \mathcal{V})^{z_2}).$$

8. Generate NIZK proofs that $com_X = (1, 1, X) \cdot \boldsymbol{f}_1^{r_{X,1}} \cdot \boldsymbol{f}_2^{r_{X,2}} \cdot \boldsymbol{f}_3^{r_{X,3}}$ and $(\Gamma_1, \Gamma_2, \Gamma_3)$ is a BBS ciphertext of $X$, as in [16]. For $\boldsymbol{f}_3 = (f_{3,1}, f_{3,2}, f_{3,3})$, we can write $com_X = (f_1^{r_{X,1}} \cdot f_{3,1}^{r_{X,3}}, f_2^{r_{X,2}} \cdot f_{3,2}^{r_{X,3}}, X \cdot g^{r_{X,1}+r_{X,2}} \cdot f_{3,3}^{r_{X,3}})$. Thus, we have

$$com_X \cdot (\Gamma_1, \Gamma_2, \Gamma_3)^{-1} = (f_1^{\chi_1} \cdot f_{3,1}^{\chi_3},\ f_2^{\chi_2} \cdot f_{3,2}^{\chi_3},\ g^{\chi_1+\chi_2} \cdot f_{3,3}^{\chi_3}), \tag{11}$$

where $\chi_1 = r_{X,1} - z_1, \chi_2 = r_{X,2} - z_2, \chi_3 = r_{X,3}$. Compute GS commitments $com_{\chi_i}$ to the exponent $\chi_i$ for $i = 1, 2, 3$ using $\tilde{\boldsymbol{f}}$, and generate the NIZK proofs $\pi_{10}, \pi_{11}, \pi_{12}$ satisfying the three linear relations (11).

9. Compute a weakly secure BB signature $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ on $\mathsf{VK}$ and the commitment $com_{\sigma_{\mathsf{VK}}}$ to $\sigma_{\mathsf{VK}}$. Next, generate the NIZK proof $\pi_{13}$ satisfying $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g, g)$.

10. Compute a one-time signature

$$\sigma_{\mathrm{OTS}} = \mathbf{Sign}_{\mathrm{OTS}}(\mathsf{SK}, (M, RL_{\tau}, \{\Gamma_i\}_{i=1}^{5}, \{\theta'_i, \tilde{\theta'}_i, \hat{\theta'}_i, \Theta'_i\}_{i=3,4,6,7}, \mathbf{com}, \mathbf{\Pi})),$$

where $\mathbf{com} = (com_{P_{\tilde{t}}}, com_{acc_{\tilde{\omega}}}, com_W, com_{J_{\tilde{t}1}}, com_{J_{\tilde{t}2}}, com_{Q_{\delta_{\tilde{k},\tilde{d}}}}, com_X, \{com_{\chi_i}\}_{i=1}^{3}, \{com_{\theta'_i}\}_{i\in\{1,2,5\}}, \{com_{\tilde{\theta'}_i}\}_{i\in\{1,2,5\}}, \{com_{\hat{\theta'}_i}\}_{i\in\{1,2,5\}},$

$\{com_{\Theta'_i}\}_{i\in\{1,2,5\}}, com_{\sigma_{\mathsf{VK}}})$, $\mathbf{\Pi} = \{\pi_i\}_{i=1}^{13}$. Output the signature $\sigma = (\mathsf{VK}, \{\Gamma_i\}_{i=1}^{5}, \{\theta'_i, \tilde{\theta'}_i, \hat{\theta'}_i, \Theta'_i\}_{i=3,4,6,7}, \mathbf{com}, \mathbf{\Pi}, \sigma_{\mathrm{OTS}})$.

**Verify.** The input are $gpk, \tau, RL_\tau, \sigma, M$. If

$$\mathbf{Verify}_{\mathrm{OTS}}(\mathsf{VK}, (M, RL_\tau, \{\Gamma_i\}_{i=1}^{5}, \{\theta'_i, \tilde{\theta'}_i, \hat{\theta'}_i, \Theta'_i\}_{i=3,4,6,7}, \mathbf{com}, \mathbf{\Pi})) = 0$$

or $\{\Gamma_i\}_{i=1}^{5}$ is not a valid tag-based encryption, output 0. Then, output 1 if all proofs are accepted. Otherwise, output 0.

**Open.** The inputs are $gpk, \tau, RL_\tau, \sigma, M, St, osk$. If **Verify** on $\sigma$ and $M$ outputs 0, output $\perp$. Otherwise, using $osk = (y_1, y_2)$, decrypt $\tilde{X} = \Gamma_3 \cdot \Gamma_1^{-1/y_1} \cdot \Gamma_2^{-1/y_2}$. Search the database $St_{trans}$ to find a record $\langle \mathsf{i}, (\mathsf{transcript}_\mathsf{i}, v, X, \{P_t, \sigma_t\}_{1\le t\le T}, sig_\mathsf{i})\rangle$ with $X = \tilde{X}$. If the search fails, output $\perp$. Otherwise, output $\mathsf{i}$.

### 4.3 Security

The proofs of the security are in the full paper.

## 5 Efficiency

We compare the efficiency of our scheme to the previous scheme [16]. In addition to parameters $N, R$, the efficiency of our system depends on $n, T, K, D$, where $T = K \cdot D$, and $n \approx T \log N$. Here, as in [16], we consider the 128-bit security level, and we assume that the element in $\mathcal{G}$ can be represented by 512 bits.

We compare the constant signature size. The signature in the previous scheme needs 144 $\mathcal{G}$-elements and the size is 9 KB. In our scheme, the signature needs 143 $\mathcal{G}$-elements, whose size is also 9 KB.

In the proposed scheme, we have the trade-off: Decreasing the revocation list size leads to increasing the sizes of public key and membership certificate. Consider the revocation list size. The revocation list consists of a non-cryptographic part related to IDs of revoked members (i.e., $\mathcal{R}_\tau, \{S_i\}_{i=1}^{m}$) and a cryptographic part of accumulators and the signatures (i.e., $\{acc_\omega, \Theta_\omega\}_{\omega=1}^{w}$). The non-cryptographic part is bounded by $5 \cdot \log N \cdot R$ bits. The cryptographic part in our scheme is bounded by $512 \cdot 8\lceil (2R-1)/T \rceil$ bits, while the part needs at most $512 \cdot 7\lceil (2R-1) \rceil$ bits in [16]. Thus, by increasing $T$, this part is greatly reduced. However, the other efficiency becomes worse as follows. The public key size of our scheme is approximately $2K \cdot T \cdot \log N \cdot 512$ bits. The membership certificate size is approximately $8 \cdot 512 \cdot T$ bits.

Next, we compare the signing costs. The computational cost of signing is comparable except for the computation of $W$. As discussed in Appendix B, $T$ exponentiations (and $2D$ exponentiations) are the extra cost compared to [16]. However, note that the computation of $W$ is required once every revocation epoch in practice. Namely, after $W$ is computed in an epoch, the following signing does not need the extra cost during the same epoch. Furthermore, we can reduce the computation of $W$ by using $W$ in the previous epoch. Thus, we consider that the extra costs are not a serious issue.

Now we consider concrete examples. We assume $N/R = 10$. To balance $K$ and $D$, we set $K = D \approx \sqrt{T}$. Table 1 shows the comparisons of the revocation list size between the previous scheme [16] and the proposed scheme using $T = 49, T = 100$, in cases of $N = 10{,}000, N = 100{,}000, N = 1{,}000{,}000$. As for the cryptographic part $(\{acc_\omega, \Theta_\omega\}_{\omega=1}^{w})$, the size is greatly reduced, as $T$ is increased. Since the non-cryptographic part cannot be reduced, we ignore cases of $T > 100$. Similarly, for $N \gg 1{,}000{,}000$, due to the huge data of the non-cryptographic part, any revocable group signatures are essentially impractical.

**Table 1.** Comparisons of the revocation list size.

| | $\mathcal{R}_\tau, \{S_i\}_{i=1}^{m}$ | $\{acc_\omega, \Theta_\omega\}_{\omega=1}^{w}$ | | |
|---|---|---|---|---|
| | | [16] | Proposed ($T = 49$) | Proposed ($T = 100$) |
| $N = 10{,}000 (R = 1{,}000)$ | 6.8 KB | 880 KB | 21 KB | 10 KB |
| $N = 100{,}000 (R = 10{,}000)$ | 83 KB | 8,800 KB | 210 KB | 100 KB |
| $N = 1{,}000{,}000 (R = 100{,}000)$ | 980 KB | 88,000 KB | 2,100 KB | 1,000 KB |

Table 2 shows the comparisons of the public key size and the membership certificate size, where $N = 1{,}000{,}000$ and $R = 100{,}000$. Since the public key size depends on only $\log N$, the size in cases of the other $N, R$ is similar to this table. The membership certificate size is the same when $N, R$ are changed. Compared to [16], the extra sizes in public key and membership certificate are needed, and are increased when $T$ is increased. In real applications, the public key and the certificate are not often distributed. On the other hand, the revocation list has to be distributed every revocation epoch. Thus, we consider that it is sufficiently practical to decrease the revocation list size while increasing the public key and the membership certificate sizes.

As for the signing cost, in our scheme, the extra cost of about 120 exponentiations is required in case of $T = 100$. The extra cost is comparable to the computations of commitments **com** with about 140 exponentiations. As shown above, the cost can be reduced in the implementation.

**Table 2.** Public key size and membership certificate size for $T$ ($N = 1{,}000{,}000, R = 100{,}000$).

| | [16] | Proposed ($T = 49$) | Proposed ($T = 100$) |
|---|---|---|---|
| Public key size ($g_{k,j}$'s) | 2.6 KB | 860 KB | 2,500 KB |
| Membership certificate size | 0.20 KB | 25 KB | 50 KB |

# A Preliminaries

## A.1 Bilinear Groups

Our scheme utilizes the following bilinear groups:

1. $\mathcal{G}$ and $\mathcal{T}$ are multiplicative cyclic groups of prime order $p$,
2. $g$ is a randomly chosen generator of $\mathcal{G}$,
3. $e$ is an efficiently computable bilinear map: $\mathcal{G} \times \mathcal{G} \to \mathcal{T}$, i.e., (1) for all $u, v \in \mathcal{G}$ and $a, b \in Z$, $e(u^a, v^b) = e(u, v)^{ab}$, and (2) $e(g, g) \neq 1_{\mathcal{T}}$.

### A.2 Assumptions

As in the underlying scheme [16], the security of our system is based on the DLIN (Decision LINear) assumption [6], the SDH (Strong DH) assumption [5], and the $q$-SFP (Simultaneous Flexible Pairing) assumption [2]. We also adopt $n$-DHE (DH Exponent) assumption [11] for the accumulator.

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

$$|\Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^{bd}, g^{c+d}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^{ac}, g^{bd}, g^z) = 1]|$$

*is negligible, where $g \in_R \mathcal{G}$ and $a, b, c, d, z \in_R Z_p$.*

**Definition 2 ($q$-SDH assumption).** *For all PPT algorithm $\mathcal{A}$ , the probability*

$$\Pr[\mathcal{A}(g, g^a, \ldots, g^{a^q}) = (b, g^{1/(a+b)}) \wedge b \in Z_p]$$

*is negligible, where $g \in_R \mathcal{G}$ and $a \in_R Z_p$.*

**Definition 3 ($q$-SFP assumption).** *For all PPT algorithm $\mathcal{A}$ , the probability*

$$\begin{aligned}\Pr[\mathcal{A}(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}, \{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^q) = (z^*, r^*, s^*, t^*, u^*, v^*, w^*) \in \mathcal{G}^7 \\ \wedge e(a, \tilde{a}) = e(g_z, z^*)e(g_r, r^*)e(s^*, t^*) \wedge e(b, \tilde{b}) = e(h_z, z^*)e(h_r, u^*)e(v^*, w^*) \\ \wedge z^* \neq 1_{\mathcal{G}} \wedge z^* \neq z_j \text{ for all } 1 \leq j \leq q]\end{aligned}$$

*is negligible, where $(g_z, h_z, g_r, h_r, a, \tilde{a}, b, \tilde{b}) \in \mathcal{G}^8$ and all tuples $\{(z_j, r_j, s_j, t_j, u_j, v_j, w_j)\}_{j=1}^q$) satisfy the above relations.*

**Definition 4 ($n$-DHE assumption).** *For all PPT algorithm $\mathcal{A}$ , the probability*

$$\Pr[\mathcal{A}(g, g^a, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}}) = g^{a^{n+1}}]$$

*is negligible, where $g \in_R \mathcal{G}$ and $a \in_R Z_p$.*

### A.3 Structure-Preserving Signatures (AHO Signatures)

We utilize the structure-preserving signatures, since the knowledge of the signature can be proved by Groth-Sahai proofs. As in [16], we adopt the AHO signature scheme in [1,2]. Using the AHO scheme, we can sign multiple group elements to obtain a constant-size signature.

**AHOKeyGen:** Select bilinear groups $\mathcal{G}, \mathcal{T}$ with a prime order $p$ and a bilinear map $e$. Select $g, G_r, H_r \in_R \mathcal{G}$, and $\mu_z, \nu_z, \mu, \nu, \alpha_a, \alpha_b \in_R Z_p$. Compute $G_z = G_r^{\mu_z}, H_z = H_r^{\nu_z}, G = G_r^{\mu}, H = H_r^{\nu}$, $A = e(G_r, g^{\alpha_a}), B = e(H_r, g^{\alpha_b})$. Output the public key as $pk = (\mathcal{G}, \mathcal{T}, p, e, g, G_r, H_r, G_z, H_z, G, H, A, B)$, and the secret key as $sk = (\alpha_a, \alpha_b, \mu_z, \nu_z, \mu, \nu)$.

**AHOSign:** Given message $M$ together with $sk$, choose $\beta, \epsilon, \eta, \iota, \kappa \in_R Z_p$, and compute $\theta_1 = g^{\beta}$, and $\theta_2 = g^{\epsilon - \mu_z \beta} M^{-\mu}$, $\theta_3 = G_r^{\eta}$, $\theta_4 = g^{(\alpha_a - \epsilon)/\eta}, \theta_5 = g^{\iota - \nu_z \beta} M^{-\nu}$, $\theta_6 = H_r^{\kappa}$, $\theta_7 = g^{(\alpha_b - \iota)/\kappa}$. Output the signature $\sigma = (\theta_1, \ldots, \theta_7)$.

**AHOVerify:** Given the message $M$ and the signature $\sigma = (\theta_1, \ldots, \theta_7)$, accept these if
$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot e(G, M)$, $B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot e(H, M)$.

This signature is existentially unforgeable against chosen-message attacks under the $q$-SFP assumption [2]. Using the re-randomization algorithm in [2], this signature can be publicly randomized to obtain another signature $(\theta'_1, \ldots, \theta'_7)$ on the same message. As a result, in the following Groth-Sahai proof, $(\theta'_i)_{i=3,4,6,7}$ can be safely revealed, while $(\theta'_i)_{i=1,2,5}$ have to be committed.

### A.4 Groth-Sahai (GS) Proofs

To prove the secrets in relations of the bilinear maps, we utilize Groth-Sahai (GS) proofs [14]. As in [16], we adopt the instantiation based on DLIN assumption. For the bilinear groups, the proof system needs a common reference string $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3) \in \mathcal{G}^3$ for $\boldsymbol{f}_1 = (f_1, 1, g), \boldsymbol{f}_2 = (1, f_2, g)$ for some $f_1, f_2 \in \mathcal{G}$. The commitment to an element $X$ is computed as $\boldsymbol{C} = (1, 1, X) \cdot \boldsymbol{f}_1^r \cdot \boldsymbol{f}_2^s \cdot \boldsymbol{f}_3^t$ for $r, s, t \in_R Z_p^*$. In case of the CRS setting for perfectly sound proofs, $\boldsymbol{f}_3 = \boldsymbol{f}_1^{\xi_1} \cdot \boldsymbol{f}_2^{\xi_2}$ for $\xi_1, \xi_2 \in_R Z_p^*$. Then, the commitment $\boldsymbol{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X g^{r+s+t(\xi_1+\xi_2)})$ is the linear encryption in [6]. On the other hand, in the setting of the witness indistinguishability, $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3$ are linearly independent, and thus $\boldsymbol{C}$ is perfectly hiding. The DLIN assumption implies the indistinguishability of the CRS.

The commitment to an exponent $x \in Z_p$ is computed as $\boldsymbol{C} = \tilde{\boldsymbol{f}}^x \cdot \boldsymbol{f}_1^r \cdot \boldsymbol{f}_2^s$ for $r, s \in_R Z_p^*$, for a CRS $\tilde{\boldsymbol{f}}, \boldsymbol{f}_1, \boldsymbol{f}_2$. In the setting of perfectly sound proofs, $\tilde{\boldsymbol{f}}, \boldsymbol{f}_1, \boldsymbol{f}_2$ are linearly independent (As in [16], for example, we can set $\tilde{\boldsymbol{f}} = \boldsymbol{f}_3 \cdot (1, 1, g)$ with $\boldsymbol{f}_3 = \boldsymbol{f}_1^{\xi_1} \cdot \boldsymbol{f}_2^{\xi_2}$). In the WI setting, $\tilde{\boldsymbol{f}} = \boldsymbol{f}_1^{\xi_1} \cdot \boldsymbol{f}_2^{\xi_2}$ provides a perfectly hiding commitment.

To prove that the committed variables satisfy the pairing relations, the prover prepares the commitments, and replaces the variables in the pairing relations by the commitments. An NIWI (non-interactive witness indistinguishable) proof allows us to prove the set of pairing product equations:

$$\prod_{i=1}^{n} e(A_i, X_i) \cdot \prod_{i=1}^{n} \prod_{j=1}^{n} e(X_i, X_j)^{a_{ij}} = t,$$

for variables $X_1, \ldots, X_n \in \mathcal{G}$ and constants $A_1, \ldots, A_n \in \mathcal{G}, a_{ij} \in Z_p, t \in \mathcal{T}$. NIWI proofs also exist for multi-exponentiation equations:

$$\prod_{i=1}^{m} A_i^{y_i} \cdot \prod_{j=1}^{n} X_j^{b_j} \cdot \prod_{i=1}^{m} \prod_{j=1}^{n} X_j^{y_i \gamma_{ij}} = T,$$

for variables $X_1, \ldots, X_n \in \mathcal{G}$, $y_1, \ldots, y_m \in Z_p$ and constants $T, A_1, \ldots, A_m \in \mathcal{G}$, $b_1, \ldots, b_n, \gamma_{ij} \in Z_p$. For the multi-exponentiation equations, we can obtain the NIZK (non-interactive zero-knowledge) proofs with no additional cost.

### A.5 Subset Cover Framework for Broadcast Encryption

As in [16], we adopt the subset cover framework for broadcast encryption in [21]. In this framework, a binary tree is used, where each leaf is assigned to each receiver (its secret key). Namely, for $N = 2^L$ receivers, the height of the tree is $L$. Let $\mathcal{N}$ be the universe of users and $\mathcal{R} \subset \mathcal{N}$ be the set of revoked receivers. In this framework, the set of non-revoked users is partitioned into $m$ disjoint subsets $S_1, \ldots, S_m$ such that $\mathcal{N} \backslash \mathcal{R} = S_1 \cup \cdots \cup S_m$.

In the framework, there are mainly the complete subtree (CS) method and the subset difference (SD) method. In the revocable group signature scheme of [16], the SD method is adapted to achieve $O(|\mathcal{R}|)$ revocation list. In this method, the disjoint set $S_i$ is determined by two nodes in the tree, *primary* node $v_{i,\phi_i}$ and *secondary* node $v_{i,\psi_i}$ that is a descendant node of $v_{i,\phi_i}$, and $S_i$ consists of the leaves of the subtree rooted by $v_{i,\phi_i}$ that are not in the subtree rooted by $v_{i,\psi_i}$. The number of subsets is bounded by $m = 2 \cdot |\mathcal{R}| - 1$, as proved in [21].

## B Evaluation of Witness Computation

In Sect. 5, the efficiency of our scheme is compared to the underlying scheme [16]. Here, we show the detailed efficiency discussion of the witness computation. The computation of $W$ can be replaced:

$$W = \prod_{1 \le d \le D} ((\prod_{i \in U} \prod_{1 \le k \le K} (\prod_{j \in V_{k,d}}^{j \neq i} g_{k,n+1-j+i}) \cdot (\prod_{j=1}^{\zeta - |V_{k,d}|, i \neq \mathsf{SP}_{k,d}} g_{k,n+1-\mathsf{SP}_{k,d}+i}))^{c_d}$$
$$\cdot (\prod_{i \in U} \prod_{1 \le k \le K} \prod_{j \in \bar{V}_{k,d}}^{j \neq i} g_{k,n+1-j+i})^{\tilde{c}_d}).$$

Then, the number of exponentiations by $c_d, \tilde{c}_d$ is $2D$. The number of multiplications is $T \cdot \log^2 N$. As discussed in [16], $\log^2 N$ multiplications is bounded by the cost of a single exponentiation. This is why $T$ exponentiations (and $2D$ exponentiations) are the extra cost compared to [16].

As mentioned in Sect. 5, the witness computation can be reduced by using $W$ in the previous epoch. In the case that the modification to the revocation

list does not influence $\mathcal{S}_{\tilde{\omega}}$ including $S_{\tilde{i}}$ (i.e., revocations happens in the other covers), the signer does not need to compute $W$. In the other cases, we can also reduce the cost: For only modified covers $S_i$ correspondent $(k, d)$, divide $W$ by the old terms for $(k, d)$ and multiply it by the new terms. Thus, we consider that the extra costs are not a serious issue.

## References


1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)
2. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133 (2010). http://eprint.iacr.org/
3. Ateniese, G., Song, D., Tsudik, G.: Quasi-efficient revocation of group signatures. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 183–197. Springer, Heidelberg (2003)
4. Begum, N., Nakanishi, T., Funabiki, N.: Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 495–509. Springer, Heidelberg (2013)
5. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security (ACM-CCS '04), pp. 168–177 (2004)
8. Bresson, E., Stern, J.: Group signature scheme with efficient revocation. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 190–206. Springer, Heidelberg (2001)
9. Camenisch, J.L., Chaabouni, R., Shelat, A.: Efficient protocols for set membership and range proofs. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 234–252. Springer, Heidelberg (2008)
10. Camenisch, J.L., Groth, J.: Group signatures: better efficiency and new theoretical aspects. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 120–133. Springer, Heidelberg (2005)
11. Camenisch, J., Kohlweiss, M., Soriente, C.: An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 481–500. Springer, Heidelberg (2009)
12. Camenisch, J.L., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
13. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
14. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

15. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
16. Libert, B., Peters, T., Yung, M.: Group signatures with almost-for-free revocation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 571–589. Springer, Heidelberg (2012)
17. Libert, B., Peters, T., Yung, M.: Scalable group signatures with revocation. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 609–627. Springer, Heidelberg (2012)
18. Libert, B., Yung, M.: Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 499–517. Springer, Heidelberg (2010)
19. Nakanishi, T., Fujii, H., Hira, Y., Funabiki, N.: Revocable group signature schemes with constant costs for signing and verifying. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 463–480. Springer, Heidelberg (2009)
20. Nakanishi, T., Funabiki, N.: Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 533–548. Springer, Heidelberg (2005)
21. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
22. Sudarsono, A., Nakanishi, T., Funabiki, N.: Efficient proofs of attributes in pairing-based anonymous credential system. In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 246–263. Springer, Heidelberg (2011)