Shortening the Libert-Peters-Yung Revocable Group Signature Scheme by Using the Random Oracle Methodology

Kazuma Ohara[*][‡], Keita Emura[§], Goichiro Hanaoka[¶], Ai Ishida[||], Kazuo Ohta[‡], and Yusuke Sakai[†][§]

[‡]The University of Electro-Communications, Japan.
[§]National Institute of Information and Communications Technology (NICT), Japan.
[¶]National Institute of Advanced Industrial Science and Technology (AIST), Japan.
[||]Tokyo Institute of Technology, Japan.

April 25, 2016

Abstract

In EUROCRYPT 2012, Libert, Peters and Yung (LPY) proposed the first scalable revocable group signature (R-GS) scheme in the standard model which achieves constant signing/verification costs and other costs regarding signers are at most logarithmic in N, where N is the maximum number of group members. However, although the LPY R-GS scheme is asymptotically quite efficient, this scheme is not sufficiently efficient in practice. For example, the signature size of the LPY scheme is roughly 10 times larger than that of the RSA signature (in 160-bit security). In this paper, we propose a compact R-GS scheme *secure in the random oracle model* that is efficient not only in the asymptotic sense but also in practical parameter settings. We achieve the same efficiency as the LPY scheme in an asymptotic sense, and the signature size is nearly equal to that of the RSA signature (in 160-bit security). It is particularly worth noting that our R-GS scheme has the *smallest* signature size compared to those of previous R-GS schemes which enable constant signing/verification costs. Our technique, which we call *parallel Boneh–Boyen–Shacham group signature technique*, helps to construct a R-GS scheme without following the technique used in LPY, i.e., we directly apply the Naor–Naor–Lotspiech framework without using any identity-based encryption.

keywords: group signature, revocation, scalability.

1 Introduction

Background: Group signature is a kind of digital signatures, proposed by Chaum and van Heyst [21]. In a group signature scheme, a group manager issues a membership certificate to each group user. Then, a signer, who has a membership certificate, can produce a group signature, and a verifier can verify whether a group signature was created by a group member or not, without identifying who the actual signer is. In order to capture a certain case, only the authority called "opener" can identify the corresponding signer of group signatures.

In many real situations, it is conceivable that signing keys will be leaked, or group members will quit. So, the revocation functionality is really desirable in practice. Currently, group signature schemes with revocation, which is called revocable group signature (R-GS), have been proposed. In particular, Libert, Peters and Yung [45] proposed the first scalable R-GS schemes, where all costs regarding signers is at most logarithmic in N, where N is the maximum number of group member. Their main technique to achieve scalability is to employ broadcast encryption, where the group manager publishes a ciphertext of a broadcast encryption scheme by indicating non-revoked members as authorized receivers, and then

[*]He currently belongs to NEC Japan.

[†]The fourth author is supported by a JSPS Fellowship for Young Scientists.

only non-revoked members can prove the decryption ability of the ciphertext. Concretely, they apply the Naor–Naor–Lotspiech framework [53] and the Dodis–Fazio construction [26], where the Complete Subtree (CS) method of the framework is implemented from identity-based encryption (IBE) and the Subset Difference (SD) method of the framework is implemented from hierarchical IBE (HIBE), respectively. LPY proposed two R-GS schemes with respect to CS and SD methods in [45]. We denote these schemes the CS-based LPY scheme and the SD-based LPY scheme, respectively.

Our Motivation: Though LPY R-GS schemes [45, 44] are asymptotically quite efficient, these schemes are not very efficient in practice. For example, the signature size of the LPY scheme [45] is roughly 10 times larger than that of the RSA signature (in 160-bit security), and that of the LPY scheme in [44] is further larger. Since the LPY R-GS schemes are constructed in the standard model and use the Groth–Sahai proof, their schemes are not efficient. Therefore, it seems natural to consider a R-GS scheme secure in the random oracle model, where the scheme realizes the same efficiency as the LPY schemes in an asymptotic sense, but has a small signature size. Moreover, from a practical perspective, even if there is an efficient scheme in the standard model, it is always meaningful to provide a more efficient scheme in the random oracle model as an alternative choice. In fact, two of the three public-key encryption schemes listed in ISO/IEC 18033-2 [1] are secure only in the random oracle model. In addition, several group signature schemes (e.g., Hwang et al. [35] or Furukawa–Imai [29]) listed in ISO/IEC 20008-2 [2] are also only provably secure in the random oracle model. We note that (R-)GS schemes secure in the standard model [33, 45, 44, 46] are constructed by Groth–Sahai proofs [34], and ones secure in the random oracle model [14, 24, 29] are constructed by the Fiat–Shamir transformation [28], which converts Σ-protocols to Non-Interactive Zero Knowledge (NIZK) proofs. Therefore, in order to construct an efficient and scalable R-GS scheme secure in the random oracle model, one may think that a LPY R-GS scheme in the random oracle model can be constructed easily via the Fiat–Shamir transformation, as in [14, 24, 29]. However, it is not straightforward to construct such a scheme due to the following two reasons.

- The languages of the Groth–Sahai proof and those of the Fiat–Shamir proof are completely different.
- There is no suitable HIBE scheme (i.e., achieving constant ciphertext size, compatibility of the Fiat–Shamir proof, etc., see below) in the random oracle model.

As for the former issue, the Groth–Sahai proofs prove pairing product equation relation, and therefore the witness of the Groth–Sahai proofs is typically "group elements". In contrast, the witness of the Fiat–Shamir proof is "discrete logarithm" of group elements, for example. Therefore, even there exists a standard model scheme which proves the possession of certain group elements, it is not obvious to directly convert the scheme to the one in the random oracle model.

As for the latter issue, the SD-based LPY scheme applies HIBE which strongly depends on the underlying algebraic structures. Moreover, the ciphertext-size must be constant in order to achieve the constant signing/verification costs, and this is the reason why the SD-based LPY scheme selects the Boneh–Boyen–Goh HIBE scheme [13]. That is, even if the Gentry–Silverberg HIBE scheme [30] (which is secure in the random oracle model) is applied, the ciphertext size is not constant and signing/verification costs depend to N. So, even if we allow to use random oracles, it seems difficult to implement the SD-based LPY scheme (and so is the LPY scheme [44] based on the concise vector commitment [48], due to the same reason) in the random oracle in an efficient way. For more discussion of the difficulty of the SD-based construction, see Section 4.4.

The last option may be the CS-based LPY scheme [45] since the CS method does not apply HIBE but IBE, and the Boneh–Franklin IBE scheme [15] is quite efficient in the random oracle model. However, the next hurdle is signatures computed by the group manager. That is, in the CS-based LPY scheme (and other LPY schemes also), the group manager computes a signature for the ciphertext of broadcast encryption for proving that the ciphertext is computed by the group manager. Moreover, since the number of ciphertexts contained in the revocation list is $O(r \log(N/r))$ in the CS-based scheme case,

Table 1: Comparison of Pairing based R-GS Schemes

Scheme	public key	signature	certificate	revocation list	signing	verification	revocation	StM/	Scalability
	size	size	size	size	cost	cost	cost	ROM[2]	
BS [16]	$O(1)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(1)$	ROM	No
NF [51]	$O(T)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	ROM	No
LV [47]	$O(T)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	StM	No
NFHF1 [50]	$O(N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	ROM	No
NFHF2 [50]	$O(\sqrt{N})$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	ROM	No
FHM [27]	$O(1)$	$O(1)$	$O(1)$	$O(N)$	$O(1)$	$O(1)$	$O(Nr)$	ROM	No
LPY1(CS) [45]	$O(1)$	$O(1)$	$O(\log N)$	$O(r \cdot \log(N/r))$	$O(1)$	$O(1)$	$O(r \cdot \log(N/r))$	StM	Yes
LPY2(SD) [45]	$O(\log N)$	$O(1)$	$O(\log^3 N)$	$O(r)$	$O(\log N)$[1]	$O(1)$	$O(r \cdot \log N)$	StM	Yes
LPY3 [44]	$O(\log N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	StM	Yes
AEHS [4]	$O(1)$	$O(1)$	$O(R)$	$O(1)$	$O(r)$[1]	$O(1)$	$O(r)$	StM	Yes
Ours	$O(1)$	$O(1)$	$O(\log N)$	$O(r \cdot \log(N/r))$	$O(1)$	$O(1)$	$O(r \cdot \log(N/r))$	ROM	Yes

N: The maximum number of group members.
T: The maximum number of revocation epochs.
r: The number of revoked users.
R: The maximum number of revoked users.
[1] This complexity is only required at the first signature generation of each revocation epoch.
[2] Standard Model / Random Oracle Model

Table 2: Comparison of Signature Length of R-GS Schemes

Scheme	$[\mathbb{G},\mathbb{Z}_p,\mathcal{F}]$[1]	80-bit security[2]	160-bit security[3]	Assumption	Pairing[5]
NFHF1 [50]	[4,16,3]	3910 bits	5888 bits	DDH on $\mathcal{F}$[1], q-SDH	Asym
NFHF2 [50]	[10,30,3]	7310 bits	11008 bits	DDH on $\mathcal{F}$[1], q-SDH	Asym
FHM [27]	[10,8,0]	3120 bits	7168 bits	DLIN, q-SDH, PDHE[4]	Sym
LPY [45]	[96,0,0]	16896 bits	49152 bits	DLIN, q-SDH	Sym
Ours	[5,13,0]	3060 bits	4608 bits	DLIN, q-SDH	Asym

[1] $\mathcal{F}$ is a group such that the Decisional Diffie–Hellman (DDH) assumption holds, different from $\mathbb{G}$.
[2] In the symmetric pairing $(|\mathbb{G}|, |\mathbb{Z}_p|, |\mathcal{F}|) = (176 \text{ bits}, 170 \text{ bits}, 170 \text{ bits})$, and in the asymmetric pairing $(|\mathbb{G}|, |\mathbb{Z}_p|, |\mathcal{F}|) = (170 \text{ bits}, 170 \text{ bits}, 170 \text{ bits})$.
[3] In the symmetric pairing $(|\mathbb{G}|, |\mathbb{Z}_p|, |\mathcal{F}|) = (512 \text{ bits}, 256 \text{ bits}, 256 \text{ bits})$, and in the asymmetric pairing $(|\mathbb{G}|, |\mathbb{Z}_p|, |\mathcal{F}|) = (256 \text{ bits}, 256 \text{ bits}, 256 \text{ bits})$.
[4] Power Diffie–Hellman Exponent assumption
[5] Symmetric Pairing/ Asymmetric Pairing

a signer needs to hide which ciphertext is selected, for achieving anonymity.[1] Therefore, even if the CS-based LPY scheme is directly constructed in the random oracle model, a signer is required to prove the same things, where the signer needs to hide the corresponding ciphertext but simultaneously needs to prove that the ciphertext is computed by the group manager, for proving the decryption ability of a broadcast encryption ciphertext. This procedure seems difficult to lead to an efficient R-GS scheme, and therefore an efficient R-GS construction in the random oracle model is still not obvious at all.

1.1 Our Contribution

In this paper, we propose the most scalable R-GS scheme secure under popular complexity assumptions (the decision linear (DLIN) assumption and the q-strong Diffie–Hellman (q-SDH) assumption) with the help of random oracle. More concretely, (1) the scheme achieves the same efficiency as the LPY schemes in an asymptotic sense, i.e., all costs regarding the signer are at most logarithmic in N, and the signing/verification costs are constant (see Table 1), and (2) the signature size of the proposed R-GS scheme is roughly 10-times smaller than that of the LPY schemes [45] in 160-bit security, which is nearly equal to that of the RSA signature scheme (See Table 2. The bit length for 80-bit security and 160-bit security is calculated by using the estimation of [22]). Our techniques are explained as follows.

[1] Recently, R-GS schemes secure in the standard model which achieve constant-size revocation list have been proposed [4, 52]. However, these schemes apply an identity-based revocation scheme [5] or extended accumulators based on [8] respectively, and these also strongly depend on the underlying algebraic structures.

1. For revocation, we *directly* apply the NNL framework without applying the use of IBE or HIBE which is used in the original LPY schemes [45]. That is, the group manager publishes a revocation list containing signatures of non-revoked users.

2. In order to prove that a signer is not revoked, the signer proves that a signature corresponding to the signer is contained in the revocation list by using the Boneh–Boyen–Shacham (BBS) group signature [14]. The construction of the proposed scheme can be seen as *parallel* BBS group signature since the possession of both (1) a membership certificate and (2) a signature contained in the revocation list are simultaneously proved by the BBS group signature scheme.

3. In order to further reduce the signature size, we apply the randomness reuse technique due to Kurosawa [40].

Our scheme is secure under the DLIN assumption and the q-SDH assumption. We additionally remark that asymmetric pairing setting is highly desirable in practice due to the recent novel works, e.g., [32, 7]. Therefore, we use the asymmetric pairing setting though the LPY schemes use the symmetric pairing setting.

Related Work: Many efficient constructions have been proposed, most of these schemes rely on the random oracle model [14, 3, 19, 37, 29, 24, 11, 55, 25]. Though most of them are based on discrete-logarithm type assumptions, Gordon et al. [31] proposed the first group signature scheme from lattice assumptions. Later, Camenisch et al. [20], Laguillaumie et al. [41], Nguyen et al. [54], and Ling et al. [49] proposed lattice-based group signature schemes with shorter group public key or signature size. Recently, Libert et al. [43] proposed a lattice-based group signature scheme with a simple joining mechanism.

Boneh, Boyen, and Shacham [14] proposed an R-GS scheme where the group manager publishes a list containing membership certificates of revoked users, and only non-revoked users can update their membership certificate from the list. This technique was also applied by Delerablée and Pointcheval [24], and Furukawa and Imai [29]. However, mainly there are two problems of this technique as follows. First, non-revoked users are involved in the revocation even when they are not revoked, and second, the membership certificate update cost is $O(r)$. In order to get rid of signers' task of the update of membership certificate, Brickell [18] proposed the concept of verifier-local revocation (VLR), where no signer is involved in the revocation procedure. In the VLR technique, a revocation list is given to verifiers and verifiers check sequentially whether the signer is included in the revocation list. This technique allows that signing cost becomes independent of the number of revoked users, but the verifying cost is $O(r)$. The specific constructions were proposed by Boneh and Shacham [16], Nakanishi and Funabiki [51], Libert and Vergnaud [47], Langlois, Ling, Nguyen and Wang [42], and so on. This VLR type technique can be used for the open functionality, and Bichsel et al. [11] constructed an efficient R-GS scheme without using encryption though the cost of opening depends on the number of users.

Though either the signing cost or the verification cost is $O(r)$ in the above methodologies, Nakanishi, Fuji, Hira, and Funabiki [50] proposed the first R-GS scheme with constant both signing and verification costs. As one drawback of their construction, the public key size is $O(\sqrt{N})$. Later, Fan, Hsu, and Manulis [27] also proposed a R-GS scheme with not only the constant signing/verification costs but also the constant size public key, however, the size of revocation list is $O(N)$.

Recently, Kumar et al. [39] proposed a group signature scheme with probabilistic revocation. In their scheme, a token (which they call alias token) is contained in a group signature, and the same token is used when group signatures are generated in the same time period, i.e., these are linkable during a time period, and this model is different from the LPY model that we adopt in this paper.

In the research of security models, first, Bellare, Micciancio, and Warinschi (BMW) [9] showed that full-anonymity and full traceability are sufficient for (static) group signatures, and now the BMW model is widely recognized as the de-facto standard security model of the group signature area. Bellare, Shi, and Zhang (BSZ) [10], and Kiayias and Yung (KY) [38, 36] independently extended the BMW model from static groups to dynamic groups. Later, Sakai et al. [58] showed that there is a room for improving the BSZ model since a signature hijacking attack is possible in the BSZ model, and proposed an extended BSZ model by considering a new security notion called opening soundness. In LPY papers [45, 44], they extended the KY model by considering the revocation functionality (the LPY model). Recently, Bootle et al. [17] pointed out that in the previous models, a user may be able to sign messages with respect to earlier time intervals during which the user was not a member of the group. Note that they also gave a countermeasure, and it is also applicable to our scheme. Since our main aim is to implement the LPY scheme in the random oracle model, we adopt the LPY model in this paper.

2 Preliminaries

In this section, we review the complexity assumptions which our scheme relies on, the BBS+ signature scheme, and the complete subtree method. Let $x \overset{R}{\leftarrow} \mathcal{X}$ denote that x as being uniformly sampled from the set $\mathcal{X}$, and $x \overset{R}{\leftarrow} X$ denote that x as being sampled from the distribution of the random variable X.

2.1 Complexity Assumptions

Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that takes a security parameter λ as input and generates a parameter $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ of bilinear groups, where p is a λ-bit prime, $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of order p, e is a bilinear map from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$, and g, h are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Here we use the asymmetric setting, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$. Similarly, we describe $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$ with the same manner.

Let $(p, \mathbb{G}, g) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$, $x \overset{R}{\leftarrow} \mathbb{Z}_p$ and $y := g^x$. The discrete logarithm (DL) problem is stated as follows: Given $(g, y, p, \mathbb{G})$, output $x = \log_g y$. The advantage of an probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ against the DL problem is defined as $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DL}}(\lambda) = \Pr[\mathcal{A}(g, y, p, \mathbb{G}) = x \mid y = g^x]$.

Definition 1 *We say that the DL assumption holds if* $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DL}}(\lambda)$ *is negligible in* λ *for any PPT algorithm* $\mathcal{A}$*.*

Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$, $\gamma \leftarrow \mathbb{Z}$ and $A_i := g^{\gamma^i}$ for $i = 0, \ldots, q$. The q-strong Diffie–Hellman (q-SDH) problem is stated as follows: Given $(g, (A_i)_{0 \leq i \leq q}, h, h^\gamma)$, output $(c, g^{1/(\gamma+c)})$ where $c \in \mathbb{Z}_p^*$. The advantage of an algorithm $\mathcal{A}$ against the q-SDH problem is defined as $\mathrm{Adv}_{\mathcal{A}}^{q\text{-SDH}}(\lambda) = \Pr[\mathcal{A}(g, (A_i)_{0 \leq i \leq q}, h, h^\gamma) = (c, g^{1/(\gamma+c)}) \wedge c \in \mathbb{Z}_p]$.

Definition 2 *We say that the q-SDH assumption holds if* $\mathrm{Adv}_{\mathcal{A}}^{q\text{-SDH}}(\lambda)$ *is negligible in* λ *for any PPT algorithm* $\mathcal{A}$*.*

Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h) \overset{R}{\leftarrow} \mathcal{G}(1^\lambda)$, u, v, $h \leftarrow \mathbb{G}_1$, α, β, $r \leftarrow \mathbb{Z}_p$ and $g_1 := u^\alpha, g_2 := v^\beta$. The decision linear (DLIN) problem is stated as follows: Given $(u, v, h, u^\alpha, v^\beta, z)$, output 1 if $z = h^{\alpha+\beta}$, otherwise 0 if $z = h^r$. The advantage of an algorithm $\mathcal{A}$ against the DLIN problem is defined as $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DLIN}}(\lambda) = |\Pr[\mathcal{A}(u, v, h, u^\alpha, v^\beta, z) = 1 \mid z = h^{\alpha+\beta}] - \Pr[\mathcal{A}(u, v, h, u^\alpha, v^\beta, z) = 1 \mid z = h^r]|$.

Definition 3 *We say that the DLIN assumption holds if* $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DLIN}}(\lambda)$ *is negligible in* λ *for any PPT algorithm* $\mathcal{A}$*.*

2.2 BBS+ Signature

We introduce the BBS+ signature scheme [29, 6] in the following. Let $g_0, g_1, \ldots, g_L, g_{L+1}$ be generators of $\mathbb{G}_1$, h be a generator of $\mathbb{G}_2$ and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing function.

Key Generation: Choose $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p^*$, and let $w = h^\gamma$. The verification key is $vk = w$, and the secret key is $sk = \gamma$.

Signing: For the message $(m_1, \ldots, m_L) \in \mathbb{Z}_p^L$, choose $\eta, \zeta \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute $A = (g_0 g_1^\zeta g_2^{m_1} \cdots g_{L+1}^{m_L})^{\frac{1}{\eta+\gamma}}$. Let the signature $\sigma = (A, \eta, \zeta)$.

Verifying: For the signature $\sigma = (A, \eta, \zeta)$ and $(m_1, \ldots, m_L)$, if $e(A, h^{\eta} vk) = e(g_0 g_1^{\zeta} g_2^{m_1} \cdots g_{L+1}^{m_L}, h)$ then output 1, and otherwise output 0.

This signature scheme has unforgeability against chosen message attack (CMA) under the q-SDH assumption. For the formal security proof, see [6]. In our usage, we set $L = 2$.

2.3 Complete Subtree Method

Naor, Naor and Lotspiech (NNL) [53] proposed the subset cover framework that is a general technique for membership revocation and traitor tracing, and this technique is used for constructing broadcast encryption. This framework is implemented by two methods: Subset Difference (SD) and Complete Subtree (CS) method. Let $\mathcal{N}$ be the set of all signers, and $\mathcal{R} \subset \mathcal{N}$ be the set of revoked signers. In such a case, the set of non-revoked users are divided into $\mathtt{num}$ disjoint sets where $\mathtt{num}$ is the number of subset. That is, $\mathcal{N} \setminus \mathcal{R} = S_1 \cup \cdots \cup S_{\mathtt{num}}$. Denote S_i $(1 \leq i \leq \mathtt{num})$ as the set of leaf nodes that have the same parent node v_i. In [53], it is proved that $\mathtt{num} \leq r \cdot \log(N/r)$ in the case of the CS method, where $\mathtt{num}$ is the number of subset and $r = |\mathcal{R}|$.

By using the CS method, we can construct a symmetric key setting broadcast encryption scheme as follows. A key is assigned to each node of a binary tree, and each user is assigned to a leaf node of the binary tree, and let $\{u_0, u_1, \ldots, u_\ell\}$ be the path from the root node to the leaf node. Then, the user obtains a key associated with each $u_j \in \{u_0, u_1, \ldots, u_\ell\}$. A ciphertext is computed by keys of nodes defined by the method. Let $\{u'_0, u'_1, \ldots, u'_{\mathtt{num}}\}$ be a set of nodes whose corresponding keys are used for encryption. If a user, whose path is $\{u_0, u_1, \ldots, u_\ell\}$, is indicated as an authorized receiver, then there exists a node u such that $u \in \{u_0, u_1, \ldots, u_\ell\} \cap \{u'_0, u'_1, \ldots, u'_{\mathtt{num}}\}$. Therefore, the user can decrypt the ciphertext using the key associated with the node u.

3 Definition of Revocable Group Signature

In this section, we give definitions of R-GS. We adopt the LPY model [45] which is a modification of the Kiayias–Yung (KY) model [38, 36].

An R-GS scheme consists of the following six probabilistic polynomial-time algorithms (Setup, Join, Revoke, Sign, Verify, Open).

Setup: It takes as inputs a security parameter $\lambda \in \mathbb{N}$ and the number of group member $N \in \mathbb{N}$, and outputs the group public key gpk, the secret key of the group manager sk_{GM}, the secret key of the opener sk_{OA}, public information represented state $St = (St_{users}, St_{trans})$. After the execution of Setup, state St is initialized as $St_{users} = \emptyset$, and $St_{trans} = \epsilon$ (empty string).

Join: It is an interactive protocol between the group manager and a signer. Let the execution of Join (that the signer takes as input λ and gpk, the group manager takes as input λ, gpk, St and sk_{GM}) denote as $[\mathsf{J}_{users}(gpk), \mathsf{J}_{GM}(gpk, St, sk_{GM})]$. By the execution of Join, the signer gets a membership certificate $cert_i$ and the secret key sec_i. In addition, St is updated as $St_{users} := St_{users} \cup \{i\}$, and $St_{trans} := St_{trans} || \langle i, \mathsf{transcript} \rangle$.

Revoke: It takes as input the set of revoked users $R_t \subset St_{users}$ for gpk, sk_{GM}, revocation epoch t, and outputs the revocation list RL_t for the epoch t.

Sign: It takes as input gpk, t, RL_t, $cert_i$, sec_i and a message M, and outputs $\perp$ if $i \in R_t$, and otherwise outputs a group signature σ.

Verify: It takes as inputs σ, t, RL_t, M, gpk, and outputs 1 if σ is valid a group signature and otherwise outputs 0.

Open: It takes as inputs M, t, RL_t, σ, sk_{OA}, gpk, St, and outputs a signer index $i \in St_{users}$ or $\perp$.

Let $cert_i \rightleftharpoons_{gpk} sec_i$ denote that $cert_i$ and sec_i are a valid certificate and a secret key by the execution of $[\mathsf{J}_{users}(gpk), \mathsf{J}_{GM}(gpk, St, sk_{GM})]$. We borrow this notation from the LPY model.

Correctness: We say that a R-GS scheme satisfy correctness when the R-GS satisfy following requirement.

1. $St = (St_{users}, St_{trans})$, where $|St_{users}| = |St_{trans}|$ (i.e., all signer is assigned to unique tag, respectively).
2. If $[\mathsf{J}_{users}(gpk), \mathsf{J}_{GM}(gpk, St, sk_{GM})]$ is executed correctly, and the signer gets $\langle i, cert_i, sec_i\rangle$, then $cert_i \rightleftharpoons_{gpk} sec_i$.
3. For all $\langle i, cert_i, sec_i\rangle$ such that $cert_i \rightleftharpoons_{gpk} sec_i$ and the revocation epoch t, the equation $\mathsf{Verify}(\sigma, M, t, RL_t, gpk) = 1$ is satisfied where $\sigma = \mathsf{Sign}(gpk, t, RL_t, cert_i, sec_i, M)$.
4. For all state St and $\langle i, cert_i, sec_i\rangle$ that issued by using St, If St' is the state that can be reached from St, For t such that $i \notin R_t$ and $\sigma = \mathsf{Sign}(gpk, t, RL_t, cert_i, sec_i, M)$, the equation $\mathsf{Open}(M, t, RL_t, \sigma, sk_{OA}, gpk, St') = i$ is satisfied.

Security Requirements: Here, we introduce the security requirements of R-GS. First, the notation and the oracles used in the definitions are given as follows:

- $state_I$: The current state. It is included $(St, gpk, sk_{GM}, sk_{OA})$ and epoch t. Initial state is $(St, gpk, sk_{GM}, sk_{OA}) \leftarrow \mathsf{Setup}(1^\lambda, N)$ and $t = 0$.
- $n = |St_{users}| < N$: The number of the group member.
- Sigs: The history of signatures issued by signing oracle. The form of each element is (i, t, M, σ), which means σ is the signature for message M in the epoch t by the signer i.
- U^a: The set of the group members that collude with the adversary.
- U^b: The set of the group members that do not collude with the adversary.
- $\mathcal{O}_{gpk}$, $\mathcal{O}_{GM}$, $\mathcal{O}_{OA}$: When these oracle are called, return gpk, sk_{GM}, sk_{OA} to the adversary, respectively.
- $\mathcal{O}_{a\text{-}join}$: The adversary executes Join with honest group manager, and the signer that collude with the adversary is added to the group. Then the number of users n is incremented and add the information of new signer to $St = (St_{users}, St_{trans})$.
- $\mathcal{O}_{b\text{-}join}$: The adversary executes Join while colluding the group manager (this signer does not collude with the adversary). Then the number of users n is incremented and add the information of new signer to $St = (St_{users}, St_{trans})$.
- $\mathcal{O}_{sig}$: It receives a query that is a message M and index i and returns $\perp$ if $i \in R_t$ or $i \notin U^b$, and otherwise returns σ for the signer i and epoch t. Then, $\mathsf{Sigs} := \mathsf{Sigs}||(i, t, M, \sigma)$.
- $\mathcal{O}_{open}$: It receives a query that is (M, σ) and epoch t, and returns the index of the signer i who generated the signature σ. Let denote $S = \{(M, \sigma, t)\}$, and we will write $\mathcal{O}_{open}^{\neg S}$ as the element of S cannot be queried to $\mathcal{O}_{open}$.
- $\mathcal{O}_{read}, \mathcal{O}_{write}$: Reading and writing $state_I$.
- $\mathcal{O}_{revoke}$: It revokes a signer from the group. It receives a query of signer index $i \in St_{users}$ and increment t, add i in R_t and update RL_t.

Next, we define anonymity, which guarantees that no adversary (who does not have sk_{OA}) can distinguish whether signers of two group signatures are the same or not.

Definition 4 (Anonymity [45]) *Anonymity is defined by the following game* $\mathrm{Exp}_{\mathcal{A}}^{anonym}$.

```
Experiment $\mathrm{Exp}_{\mathcal{A}}^{anonym}(\lambda)$
    $state_I = (St, gpk, sk_{GM}, sk_{OA}) \leftarrow \mathsf{Setup}(1^\lambda, N)$
    $(aux, M^*, t^*, RL_{t^*}, (sec_i^*, cert_i^*), (sec_1^*, cert_1^*)) \leftarrow \mathcal{A}(play : \mathcal{O}_{gpk}, \mathcal{O}_{GM}, \mathcal{O}_{revoke}, \mathcal{O}_{open}, \mathcal{O}_{read}, \mathcal{O}_{write})$
    If $\neg cert_b^* \rightleftharpoons_{gpk} sec_b^*$ or $\mathsf{IsRevoked}(sec_b^*, cert_b^*, RL_{t^*}) = 1$ for $b \in \{0,1\}$ or $cert_0^* = cert_1^*$ then return $0$
    $d \xleftarrow{R} \{0,1\}$; $\sigma^* \leftarrow \mathsf{Sign}(gpk, t^*, cert_d^*, sec_d^*, M^*)$
    $d' \leftarrow \mathcal{A}(guess : \sigma^*, aux, \mathcal{O}_{gpk}, \mathcal{O}_{GM}, \mathcal{O}_{open}^{\neg\{(M^*,\sigma^*,t^*)\}}, \mathcal{O}_{read}, \mathcal{O}_{write})$
    If $d' = d$ then return $1$
    Return $0$.
```

The advantage of the adversary $\mathcal{A}$ *against the above game is* $\mathrm{Adv}_{\mathcal{A}}^{anonym}(\lambda) = |\Pr[\mathrm{Exp}_{\mathcal{A}}^{anonym}(\lambda) = 1] - 1/2|$. *We say that the R-GS scheme satisfies anonymity if* $\mathrm{Adv}_{\mathcal{A}}^{anonym}(\lambda)$ *is negligible in* λ *for any probabilistic polynomial-time algorithm* $\mathcal{A}$.

Next, we define non-frameability which guarantees that no adversary (who can corrupt the group manager and the opener) can produce a group signature whose opening result is an honest user.

Definition 5 (Non-Frameability [45]) *Non-frameability is defined by the following game* $\mathrm{Exp}_{\mathcal{A}}^{frame}$.

```
Experiment $\mathrm{Exp}_{\mathcal{A}}^{frame}(\lambda)$
    $state_I = (St, gpk, sk_{GM}, sk_{OA}) \leftarrow \mathsf{Setup}(1^\lambda, N)$
    $(M^*, \sigma^*, t^*, RL_{t^*}) \leftarrow \mathcal{A}(\mathcal{O}_{gpk}, \mathcal{O}_{GM}, \mathcal{O}_{OA}, \mathcal{O}_{b\text{-}join}, \mathcal{O}_{revoke}, \mathcal{O}_{sig}, \mathcal{O}_{read}, \mathcal{O}_{write})$
    If $\mathsf{Verify}(\sigma^*, M^*, t^*, RL_{t^*}, gpk) = 0$ then return $0$
    $i = \mathsf{Open}(M^*, t^*, RL_{t^*}, \sigma^*, sk_{OA}, gpk, St')$
    If $i \notin U^b$ return $0$
    If $(\wedge_{j \in U^b s.t. j=i}(j, t^*, M^*, *) \notin \mathsf{Sigs})$ then return $1$
    Return $0$.
```

The advantage of $\mathcal{A}$ *against the above game is* $\mathrm{Adv}_{\mathcal{A}}^{frame}(\lambda) = \Pr[\mathrm{Exp}_{\mathcal{A}}^{frame}(\lambda) = 1]$. *We say that the R-GS scheme satisfies non-frameability if* $\mathrm{Adv}_{\mathcal{A}}^{frame}(\lambda)$ *is negligible in* λ *for any probabilistic polynomial-time algorithm* $\mathcal{A}$.

Next, we define misidentification resistance which guarantees that no adversary (who does not have sk_{GM}) can produce a valid group signature whose opening result is in outside of the set of non-revoked adversarially-controlled users.

Definition 6 (Misidentification resistance [45]) *Misidentification resistance is defined by the following game* $\mathrm{Exp}_{\mathcal{A}}^{misid}$.

```
Experiment $\mathrm{Exp}_{\mathcal{A}}^{misid}(\lambda)$
    $state_I = (St, gpk, sk_{GM}, sk_{OA}) \leftarrow \mathsf{Setup}(1^\lambda, N)$
    $(M^*, \sigma^*, t^*, RL_{t^*}) \leftarrow \mathcal{A}(\mathcal{O}_{gpk}, \mathcal{O}_{a\text{-}join}, \mathcal{O}_{revoke}, \mathcal{O}_{read}, \mathcal{O}_{OA})$
    If $\mathsf{Verify}(\sigma^*, M^*, t^*, RL_{t^*}, gpk) = 0$ then return $0$
    $i = \mathsf{Open}(M^*, t^*, RL_{t^*}, \sigma^*, sk_{OA}, gpk, St')$
    If $(i \notin U^a \setminus R_{t^*})$ return $1$
    Return $0$.
```

The advantage of $\mathcal{A}$ *against the above game is* $\mathrm{Adv}_{\mathcal{A}}^{misid}(\lambda) = \Pr[\mathrm{Exp}_{\mathcal{A}}^{misid}(\lambda) = 1]$. *We say that the R-GS scheme satisfies misidentification resistance if* $\mathrm{Adv}_{\mathcal{A}}^{misid}(\lambda)$ *is negligible in* λ *for any probabilistic polynomial-time algorithm* $\mathcal{A}$.

4 Proposed R-GS Scheme

In this section, we give the proposed R-GS scheme. First, we explain our technique called parallel BBS group signature technique which is the core technique of our R-GS construction.

4.1 An NIZK Proof for Parallel BBS Group Signature Technique

In our R-GS scheme, each signer is associated to a leaf node of a binary tree. Let $g, g_1, g_2, f_1, f_2, f_3, h_0, h_1, X \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. Let $\{u_0, u_1, \dots, u_\ell\}$ be the path from the root node to the leaf node. Then, the signer is issued BBS+ signatures $\{A_j = (gh_0^{\zeta_j} h_1^{u_j} X)^{\frac{1}{\gamma_0 + \eta_j}}\}_{j \in [1,\ell]}$ for all $u_j \in \{u_0, u_1, \dots, u_\ell\}$ as the membership certificate. A revocation list contains BBS+ signatures $\{B_{i,t} = (gh_0^{\zeta'_i} h_1^{u_i} h_2^t)^{\frac{1}{\eta'_i + \gamma_1}}\}$ for all $u_i \in \{u'_0, u'_1, \dots, u'_{\mathtt{num}}\}$ where $\{u'_0, u'_1, \dots, u'_{\mathtt{num}}\}$ is determined by the CS method. If a signer is not revoked, then there exist two signatures A_j and $B_{i,t}$ that sign the same node $u_j = u_i$.

In order to describe our R-GS scheme, first, we show NIZK proofs which prove the possession of two BBS+ signatures and also prove the equality of the two signed messages. Since we use two BBS group signatures simultaneously, we call it the parallel BBS group signature technique. The fact to be proved is described as follows:

- A signer i has a membership certificate A_j that proves "the signer belongs to the group." Let u_j be the signed message of A_j where $u_j \in \{u_0, u_1, \dots, u_\ell\}$.
- A signature $B_{j,t}$, whose signed message is also u_j, is contained in the revocation list that proves "the signer who is a descendant of the node u_j is not revoked at time t."
- A_j held by the signer contains a secret key x, which is hidden against even the group manager (for non-frameability).

We prove the above statement as follows: Let $\theta = (A, \eta, \zeta)$ be a BBS+ signature for the message (m, x) such that $A = (gh_0^{\zeta} h_1^m h_2^x)^{\frac{1}{\eta + \gamma_0}}$. Let $\Theta = (B_t, \eta', \zeta')$ be a BBS+ signature for the message (m', t) such that $B_t = (gh_0^{\zeta'} h_1^{m'} h_2^t)^{\frac{1}{\eta' + \gamma_1}}$. The statement of the protocol, that proves that possession of (A, x) and B_t such that $m = m'$, is described as follows. Let $vk_0 = h^{\gamma_0}$ and $vk_1 = h^{\gamma_1}$ be the verification key for the BBS+ signatures A and B_t with $m = m'$, respectively, and H be a random oracle. The prover chooses $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p$ and encrypts A and B_t as follows:

$$\psi_1 = f_1^{\alpha},\ \psi_2 = f_2^{\beta},\ \psi_3 = f_3^{\alpha+\beta}, \psi_4 = (g_1^{\alpha} g_2^{\beta} A),\ \psi_5 = (g_1'^{\alpha} g_2'^{\beta} B_t).$$

As a remark, the prover needs to encrypt both A and B_t since information of which A and B_t are used helps an adversary to break anonymity in the proposed group signature scheme.

Moreover, we note that A and B_t are encrypted using same scheme and its randomnesses are reused. This technique comes from Kurosawa's multi-recipient public-key encryption. This paper shows that randomness in the Cramer–Shoup encryption [23] can be reused for directing different messages to different recipients. We use this technique in order to encrypt a vector of messages. Intuitively, this might be reminiscent of the fact that under the decisional Diffie–Hellman assumption, given g^x, g^{y_1}, and g^{y_2} the two elements g^{xy_1} and g^{xy_2} look random elements. For further details, please refer to our security proof or the [40] paper.

Next, the prover proves that the value $(\alpha, \beta, x, m, \eta, \eta', \zeta, \zeta')$ satisfies the following relation, where $\eta, \eta' \xleftarrow{R} \mathbb{Z}_p$:

$$\begin{aligned}
&\psi_1 = f_1^{\alpha},\ \psi_2 = f_2^{\beta},\ \psi_3 = f_3^{\alpha+\beta}, \\
&e(\psi_4 \cdot g_1^{-\alpha} g_2^{-\beta}, h^{\eta} vk_0) = e(gh_0^{\zeta} h_1^m h_2^x, h),\ \psi_1^{\eta} f_1^{-\alpha\eta} = 1,\ \psi_2^{\eta} f_2^{-\beta\eta} = 1, \\
&e(\psi_5 \cdot g_1'^{-\alpha} g_2'^{-\beta}, h^{\eta'} vk_1) = e(gh_0^{\zeta'} h_1^m h_2^t, h),\ \psi_1^{\eta'} f_1^{-\alpha\eta'} = 1,\ \psi_2^{\eta'} f_2^{-\beta\eta'} = 1.
\end{aligned}$$

The first line and second line, and the first line and third line are the statement of the BBS group signature scheme. This relations can be seen as parallel BBS group signature for A (the first line and second line) and B_t (the first line and third line), respectively, where α and β are reused. As a remark, the equation $\psi_1^{\eta} f_1^{-\alpha\eta} = 1$ $(resp. \psi_1^{\eta'} f_1^{-\alpha\eta'} = 1)$ is for proving the validity of $\alpha\eta$ (resp. $\alpha\eta'$).[2] Note that the value t is not a witness, since t (which indicates a revocation epoch in our R-GS scheme) is a public value.

Here, we give the NIZK proof which is constructed from a Σ-protocol for proving $(\alpha, \beta, x, m, \eta, \eta', \zeta, \zeta')$ via the Fiat–Shamir transformation. Briefly, random values are chosen for each witness and R values are computed according to the relation to be proved. Note that the suffix appeared in random/R values indicate the corresponding witness.

Proof: The proof of the above relations is as follows: Choose $r_\alpha, r_\beta, r_\eta, r_\zeta, r_{\eta'}, r_{\zeta'}, r_{\alpha\eta}, r_{\beta\eta}, r_{\alpha\eta'}, r_{\beta\eta'}, r_m, r_x \xleftarrow{R} \mathbb{Z}_p$ and compute $R_\alpha, R_\beta, R_{\alpha+\beta}, R_A, R_B, R_{\alpha\eta}, R_{\beta\eta}, R_{\alpha\eta'}, R_{\beta\eta'}$ in the following.

$$
\begin{aligned}
&R_\alpha \leftarrow f_1^{r_\alpha},\ R_\beta \leftarrow f_2^{r_\alpha},\ R_{\alpha+\beta} \leftarrow f_3^{r_\alpha + r_\beta},\\
&R_A \leftarrow e(\psi_4, h)^{r_\eta} e(g_1, h)^{-r_{\alpha\eta}} e(g_1, vk_0)^{-r_\alpha} e(g_2, h)^{-r_{\beta\eta}} e(g_2, vk_0)^{-r_\beta} e(h_0, h)^{-r_\zeta} e(h_1, h)^{-r_m} e(h_2, h)^{-r_x},\\
&R_{\alpha\eta} \leftarrow \psi_1^{r_\eta} f_1^{-r_{\alpha\eta}},\ R_{\beta\eta} \leftarrow \psi_2^{r_\eta} f_2^{-r_{\beta\eta}},\\
&R_B \leftarrow e(\psi_5, h)^{r_{\eta'}} e(g'_1, h)^{-r_{\alpha\eta'}} e(g'_1, vk_0)^{-r_\alpha} e(g'_2, h)^{-r_{\beta\eta'}} e(g'_2, vk_0)^{-r_\beta} e(h_0, h)^{-r_{\zeta'}} e(h_1, h)^{-r_m},\\
&R_{\alpha\eta'} \leftarrow \psi_1^{r_{\eta'}} f_1^{-r_{\alpha\eta'}},\ R_{\beta\eta'} \leftarrow \psi_2^{r_{\eta'}} f_2^{-r_{\beta\eta'}}.
\end{aligned}
$$

Here, $(R_\alpha, R_\beta, R_{\alpha+\beta}, R_A, R_{\alpha\eta}, R_{\beta\eta})$ and $(R_\alpha, R_\beta, R_{\alpha+\beta}, R_B, R_{\alpha\eta'}, R_{\beta\eta'})$ correspond to a BBS group signature, respectively. Again, R_α, R_β, and $R_{\alpha+\beta}$ are commonly used. Using the above $R_\alpha, \ldots, R_{\beta\eta'}$, compute

$$c \leftarrow H(\psi_1, \ldots, \psi_5, R_\alpha, \ldots, R_{\beta\eta'}),$$

and then compute following values:

$$
\begin{aligned}
&s_\alpha \leftarrow r_\alpha + c\alpha,\ s_\beta \leftarrow r_\beta + c\beta,\ s_\eta \leftarrow r_\eta + c\eta,\ s_\zeta \leftarrow r_\zeta + c\zeta,\ s_{\eta'} \leftarrow r_{\eta'} + c\eta',\\
&s_{\zeta'} \leftarrow r_{\zeta'} + c\zeta',\ s_{\alpha\eta} \leftarrow r_{\alpha\eta} + c\alpha\eta,\ s_{\alpha\eta'} \leftarrow r_{\alpha\eta'} + c\alpha\eta',\ s_{\beta\eta} \leftarrow r_{\beta\eta} + c\beta\eta,\\
&s_{\beta\eta'} \leftarrow r_{\beta\eta'} + c\beta\eta',\ s_m \leftarrow r_m + cm,\ s_x \leftarrow r_x + cx.
\end{aligned}
$$

Finally, output the proof $\pi = (c, s_\alpha, s_\beta, s_\eta, s_\zeta, s_{\eta'}, s_{\zeta'}, s_{\alpha\eta}, s_{\alpha\eta'}, s_{\beta\eta}, s_{\beta\eta'}, s_m, s_x)$, and the prover sends $(\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \pi)$ to the verifier.

Verify: The verifier computes the following values from π:

$$
\begin{aligned}
&R'_\alpha \leftarrow f_1^{s_\alpha} \psi_1^{-c},\ R'_\beta \leftarrow f_2^{s_\alpha} \psi_2^{-c},\ R'_{\alpha+\beta} \leftarrow f_3^{s_\alpha + s_\beta} \psi_3^{-c},\\
&R'_A \leftarrow e(\psi_4, h)^{s_\eta} e(g_1, h)^{-s_{\alpha\eta}} e(g_1, vk_0)^{-s_\alpha} e(g_2, h)^{-s_{\beta\eta}} e(g_2, vk_0)^{-s_\beta} e(h_0, h)^{-s_\zeta}\\
&\qquad \cdot e(h_1, h)^{-s_m} e(h_2, h)^{-s_x} (e(g, h)/e(\psi_4, vk_0))^{-c},\\
&R'_{\alpha\eta} \leftarrow \psi_1^{s_\eta} f_1^{-s_{\alpha\eta}},\ R'_{\beta\eta} \leftarrow \psi_2^{s_\eta} f_2^{-s_{\beta\eta}},\\
&R'_B \leftarrow e(\psi_5, h)^{s_{\eta'}} e(g'_1, h)^{-s_{\alpha\eta'}} e(g'_1, vk_0)^{-s_\alpha} e(g'_2, h)^{-s_{\beta\eta'}} e(g'_2, vk_0)^{-s_\beta} e(h_0, h)^{-s_{\zeta'}}\\
&\qquad \cdot e(h_1, h)^{-s_m} (e(g, h) e(h_2, h)^t / e(\psi_5, vk_1))^{-c},\\
&R'_{\alpha\eta'} \leftarrow \psi_1^{s_{\eta'}} f_1^{-s_{\alpha\eta'}},\ R'_{\beta\eta'} \leftarrow \psi_2^{s_{\eta'}} f_2^{-s_{\beta\eta'}}.
\end{aligned}
$$

If $c = H(\psi_1, \ldots, \psi_5, R'_\alpha, \ldots, R'_{\beta\eta'})$ then the proof π is accepted, and otherwise rejected.

In the next section, we give the proposed R-GS scheme. In our R-GS scheme, A (generated in the Join algorithm) is a membership certificate and B (generated in the Revoke algorithm and contained in the revocation list) is a signature corresponding to non-revoked users.

[2] Note that in the original BBS group signature scheme a Boneh–Boyen (BB) short signature [12] is used as a membership certificate. Since we need to use a signature scheme with multiple signed messages, we use the BBS+ signature scheme instead of the BB signature scheme.

4.2 The Proposed R-GS Construction via Parallel BBS Group Signature Technique

Here, the construction of the proposed scheme is described as follows. In our scheme, a signer is assigned to a leaf node, and let $(u_0, u_1, \ldots, u_\ell)$ be the path from the root to the leaf. Then the signer is issued a membership certificate according to the path such that $\{A_j = (gh_0^{\zeta_j} h_1^{u_j} X)^{\frac{1}{\gamma_0+\eta_j}}\}_{j\in[1,\ell]}$. Moreover, if the signer is not revoked (at t), $B_{i,t} = (gh_0^{\zeta'_i} h_1^{u_i} h_2^t)^{\frac{1}{\eta'_i+\gamma_1}}$ is contained in the revocation list where $u_i = u_j$. Then, a group signature is computed shown in Sect. 4.1 by setting $m = u_j$ as a signed message of the BBS+ signature. Note that x (chosen in the Join algorithm) is known by a user only, and therefore no group manager can make a group signature instead of the user.

$\mathsf{Setup}(1^\lambda, N)$**:** Choose $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h) \xleftarrow{R} \mathcal{G}(1^\lambda)$, $f_1, f_2, f_3, h_0, h_1, h_2 \xleftarrow{R} \mathbb{G}_1 \setminus \{1\}$. Let $\gamma_0, \gamma_1 \xleftarrow{R} \mathbb{Z}_p$ and $(sk_0, vk_0) = (\gamma_0, h^{\gamma_0})$, $(sk_1, vk_1) = (\gamma_1, h^{\gamma_1})$. Then, choose $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3 \xleftarrow{R} \mathbb{Z}_p$, and compute $g_1 = f_1^{\xi_1} f_3^{\xi_3}, g_2 = f_2^{\xi_2} f_3^{\xi_3}, g'_1 = f_1^{\xi'_1} f_3^{\xi'_3}, g'_2 = f_2^{\xi'_2} f_3^{\xi'_3}$. Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. Let $sk_{OA} = (\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3)$, $sk_{GM} = (sk_0, sk_1) = (\gamma_0, \gamma_1)$, $gpk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, f_1, f_2, f_3, g_1, g_2, g'_1, g'_2, h_0, h_1, h_2, h, vk_0, vk_1, H)$, and $St = (St_{users}, St_{trans}) = (\emptyset, \epsilon)$. Finally, output $sk_{OA}, sk_{GM}, gpk, St$.

Join**:** A user i chooses $x \xleftarrow{R} \mathbb{Z}_p$ and computes a signature sig_i for the message $X = h_2^x$, then send (X, sig_i) to the group manager. Next, the group manager assigns the user i to a leaf u_ℓ of the binary tree. Let $u_0, u_1, \ldots, u_\ell$ be the path from the root node to the lead node. For $j = 0, \ldots, \ell$, the group manager chooses $\eta_j, \zeta_j \xleftarrow{R} \mathbb{Z}_p$, and computes $A_j = (gh_0^{\zeta_j} h_1^{u_j} X)^{\frac{1}{\gamma_0+\eta_j}}$. Then, the group manager sends $\{\theta_j = (A_j, \eta_j, \zeta_j)\}_{j=0}^{\ell}$ and $\langle v_i \rangle := (u_0, \ldots, u_\ell)$ to the user i. The user obains the user membership certificate $cert_i = (\langle v_i \rangle, \{A_j\}_{j=0}^{\ell}, X)$ and secret key $sec_i = x$, respectively. Finally, the group manager adds i and $\mathsf{transcript}_i = (X, \{A_j\}_{j=0}^{\ell}, sig_i)$ to the state St_{trans}.

$\mathsf{Revoke}(gpk, sk_{GM}, t, R_t)$**:** Determine the set of node $\{u'_0, u'_1, \ldots, u'_{\mathtt{num}}\}$ from the CS method (note that $\mathtt{num} \le r \cdot \log(N/r)$). For all i, choose $\eta'_i, \zeta'_i \xleftarrow{R} \mathbb{Z}_p$, then compute $B_{i,t} = (gh_0^{\zeta'_i} h_1^{u_i} h_2^t)^{\frac{1}{\eta'_i+\gamma_1}}$ and let $\{\Theta = (B_{i,t}, \eta'_i, \zeta'_i)\}_{i=0}^{\mathtt{num}}$. Then, output $RL_t = (t, R_t, \{\Theta_i\}_{i=1}^{\mathtt{num}})$.

$\mathsf{Sign}(gpk, t, RL_t, cert_i, sec_i, M)$**:** If $i \in R_t$ then return $\perp$. Otherwise, the signature is computed as follows. Since $i \notin R_t$, there exist (A_j, x) and $B_{j,t}$ such that $A_j = (gh_0^{\zeta_j} h_1^{u_j} h_2^x)^{\frac{1}{\gamma_0+\eta_j}}$ and $B_{j,t} = (gh_0^{\zeta'_i} h_1^{u_j} h_2^t)^{\frac{1}{\eta'_i+\gamma_1}}$. Choose $\alpha, \beta \xleftarrow{R} \mathbb{Z}_p$ and compute $\psi_1 = f_1^\alpha$, $\psi_2 = f_2^\beta$, $\psi_3 = f_3^{\alpha+\beta}$, $\psi_4 = (g_1^\alpha g_2^\beta A_j)$, and $\psi_5 = (g_1'^\alpha g_2'^\beta B_{j,t})$. Then, the signer issues an NIZK proof π that proves (A_j, x) is possessed and u_j corresponds to both A_j and $B_{j,t}$ as shown in Sect. 4.1 by setting $m := u_j$. Again note that t is not a witness. Moreover, we note that the signed message M is included such as $c \leftarrow H(M, \psi_1, \ldots, \psi_5, R_\alpha, \ldots, R_{\beta\eta'})$. Finally, output the signature $\sigma = (\psi_1, \ldots, \psi_5, \pi)$.

$\mathsf{Verify}(\sigma, M, t, RL_t, gpk)$**:** The verifier checks the NIZK proof π as the verifying procedure in Sect. 4.1. Note that the signed message M is included such as $c = H(M, \psi_1, \ldots, \psi_5, R'_\alpha, \ldots, R'_{\beta\eta'})$. If π is accepted, then output 1, and otherwise output 0.

$\mathsf{Open}(M, t, RL_t, \sigma, sk_{OA}, gpk, St)$**:** Compute $A' = (\psi_4/\psi_1^{\xi_1}\psi_2^{\xi_2}\psi_3^{\xi_3})$. If there exists $\langle i, \mathsf{transcript}_i \rangle = (X, \{\theta_j\}_{j=0}^{\ell}, sig_i)$ where $\theta = (A', *, *)$ in St_{trans}, then verify sig_i and output i if sig_i is a valid signature, and otherwise output $\perp$.

4.3 Security

The proposed scheme satisfies the following Theorems 4.1, 4.2, and 4.3.

Theorem 4.1 *The proposed R-GS scheme has anonymity in the random oracle model under the DLIN assumption, where H is modeled as a random oracle.*

Since anonymity means that no signer can be identified without opener's secret key, the attack on anonymity is equal to the attack on the encryption $(\psi_1, \ldots, \psi_5)$ by which membership certificate is encrypted.Namely, the anonymity of this scheme is reduced to the CCA security of the linear encryption scheme [59].

The concrete proof is given as follows. The proof proceeds with a sequence of games. First, we define the following games. In the following we denote by S_i the event that in Game i the adversary successfully guesses the bit picked by the challenger.

Game 0. The initial game is identical to the game defined in the definition of admitter anonymity. We assumed that queries to the hash function are responded by the challenger. For this purpose the challenger maintains a hash list, which contains tuples of the form $(M, \psi_1, \ldots, \psi_5, R_\alpha, \ldots, R_{\beta\eta'}, c)$. for the hash function H.

Game 1. In this game we replace the zero-knowledge proof of the challenge signature with a simulated proof. When the adversary asks a challenge signature $(\psi_1^*, \ldots, \psi_5^*, c^*, R_\alpha^*, \ldots, R_{\beta\eta'}^*)$ by sending (i_0, i_1, M), the challenger computes it as follows]: the challenger flips the bit $b \in \{0,1\}$, computes $(\psi_1^*, \ldots, \psi_5^*)$ as specified in the construction with the signing key $(cert_{i_b}, sec_{i_b})$, generates random integers $c^*, s_\alpha^*, \ldots, s_{\beta\eta'}^* \leftarrow \mathbb{Z}_p^*$, and compute

$$\begin{aligned}
&R'_\alpha \leftarrow f_1^{s_\alpha^*} \psi_1^{*-c^*},\ R'_\beta \leftarrow f_2^{s_\alpha^*} \psi_2^{*-c^*},\ R'_{\alpha+\beta} \leftarrow f_3^{s_\alpha^* + s_\beta^*} \psi_3^{*-c^*}, \\
&R'_A \leftarrow e(\psi_4^*, h)^{s_\eta^*} e(g_1, h)^{-s_{\alpha\eta}^*} e(g_1, vk_0)^{-s_\alpha^*} e(g_2, h)^{-s_{\beta\eta}^*} e(g_2, vk_0)^{-s_\beta^*} e(h_0, h)^{-s_\zeta^*} \\
&\qquad \cdot e(h_1, h)^{-s_m^*} e(h_2, h)^{-s_x^*} (e(g, h)/e(\psi_4^*, vk_0))^{-c^*}, \\
&R'_{\alpha\eta} \leftarrow \psi_1^{*s_\eta^*} f_1^{-s_{\alpha\eta}^*},\ R'_{\beta\eta} \leftarrow \psi_2^{*s_\eta^*} f_2^{-s_{\beta\eta}^*}, \\
&R'_B \leftarrow e(\psi_5^*, h)^{s_{\eta'}^*} e(g'_1, h)^{-s_{\alpha\eta'}^*} e(g'_1, vk_0)^{-s_\alpha^*} e(g'_2, h)^{-s_{\beta\eta'}^*} e(g'_2, vk_0)^{-s_\beta^*} e(h_0, h)^{-s_{\zeta'}^*} \\
&\qquad \cdot e(h_1, h)^{-s_m^*} (e(g, h) e(h_2, h)^t / e(\psi_5^*, vk_1))^{-c^*}, \\
&R'_{\alpha\eta'} \leftarrow \psi_1^{*s_{\eta'}^*} f_1^{-s_{\alpha\eta'}^*},\ R'_{\beta\eta'} \leftarrow \psi_2^{*s_{\eta'}^*} f_2^{-s_{\beta\eta'}^*}.
\end{aligned}$$

The challenger adds the tuple $(M, \psi_1^*, \ldots, \psi_5^*, R_\alpha^*, \ldots, R_{\beta\eta'}^*, c^*)$ to the hash list for H. At this point if the list for H already contains a tuple of the form $(M, \psi_1^*, \ldots, \psi_5^*, R_\alpha^*, \ldots, R_{\beta\eta'}^*, c)$ for some c, the challenger outputs $\perp$ and halts. Otherwise the challenger sends $(\psi_1^*, \ldots, \psi_5^*, c^*, s_\alpha^*, \ldots, s_{\beta\eta'}^*)$ to the adversary as the challenge signature. We will argue that this change introduce only a negligible difference in the adversary's advantage.

Game 2. In this game we modify the linear encryption in the challenge to be "invalid." More precisely, to compute the challenge $(\psi_1^*, \ldots, \psi_5^*, c^*, s_\alpha^*, \ldots, s_{\beta\eta'}^*)$, the challenger selects random integers $\alpha, \beta \leftarrow \mathbb{Z}_p$ and $\tau \leftarrow \mathbb{Z}_p \setminus \{\alpha + \beta\}$, and computes $\psi_1^* = f_1^\alpha$, $\psi_2^* = f_2^\beta$, $\psi_3^* = f_3^\tau$, $\psi_4^* = (\psi_1^*)^{\xi_1} (\psi_2^*)^{\xi_2} (\psi_3^*)^{\xi_3} A_{i_b}$, and $\psi_5^* = (\psi_1^*)^{\xi'_1} (\psi_2^*)^{\xi'_2} (\psi_3^*)^{\xi'_3} B_{i_b}$ where f_1, f_2, f_3 are the part of the group public key gpk, b is the bit flipped for the challenge, A_{i_b} is the part of the user membership certificate of the member i_b, B_{i_b} is the signature of the group manager in the revocation list RL_{t^*} corresponding the member i_b. Notice that challenger uses the secret key sk_{OM} for opener (actually its component $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$) to compute the challenge. All the other components of the challenge is generated as in Game 1. This modification also does not change the adversary's winning probability non-negligibly, provided that the DLIN assumption holds.

Game 3. In this game we modify the opening oracle to reject a signature $(\psi_1, \ldots, \psi_5, s_\alpha, \ldots, s_{\beta\eta'})$ when it satisfies the following two conditions: $(\psi_1, \ldots, \psi_5) = (\psi_1^*, \ldots, \psi_5^*)$, that is, the component $\psi_1, \ldots, \psi_5$ in the query are reused from the challenge signature, and $(R'_\alpha, \ldots, R'_{\beta\eta'}) = (R_\alpha^*, \ldots, R_{\beta\eta'}^*)$, where $(R'_\alpha, \ldots, R'_{\beta\eta\prime})$ is the group elements reproduced in the verification process. This change does not affect the adversary's advantage non-negligibly.

Game 4. We further introduce another rejection rule. In this game the opening oracle rejects a signature that contains a ciphertext whose linear encryption component (ψ_1, ψ_2, ψ_3) does not constitute a linear tuple. Specifically when ψ_1, ψ_2, ψ_3 satisfy $\psi_1 = f_1^{\alpha}$, $\psi_2 = f_2^{\beta}$, $\psi_3 = f_3^{v}$, the challenger immediately rejects queries such that $\alpha + \beta \neq v$, and all other queries are treated as before. This modification does not affect the behavior of the adversary, as the adversary can issue such as invalid query with a valid (that passes the verification) proof only with negligible probability.

Lemma 4.1 $|\Pr[S_0] - \Pr[S_1]|$ *is negligible.*

Proof. We claim that the distribution (of the challenge) in Game 1 is identical to that in Game 0 except for cases in which the challenger outputs $\perp$. This follows from a standard discussion of the simulation of zero-knowledge proof. To see this, we can observe that $s_{\alpha}^{*} - c^{*}\alpha$ in Game 1 corresponding to r_{α} in Game 0, and similar correspondence holds for all other s^{*}'s and r's. We can also see that both $s_{\alpha}^{*} - c^{*}\alpha$ and r_{α} are uniformly distributed over $\mathbb{Z}_p$. We will then see that the challenger in Game 1 outputs $\perp$ only with negligible probability. It can be obtained from the fact that $(R_{\alpha}^{*}, \ldots, R_{\beta\eta'}^{*})$ are distributed uniformly over a set with cardinality (at least) p, that is, the oracle queries to H issued before the challenge phase contain $(M, \psi_1^{*}, \ldots, \psi_5^{*}, R_{\alpha}^{*}, \ldots, R_{\beta\eta'}^{*}, c)$ with probability (at most) q_H/p where q_H denotes the number of oracle queries to H issued by the adversary. □

Lemma 4.2 $|\Pr[S_1] - \Pr[S_2]|$ *is negligible, provided that the DLIN assumption holds.*

Proof. We will describe a distinguishing algorithm $\mathcal{B}$ of the DLIN problem to bound the absolute difference $|\Pr[S_1] - \Pr[S_2]|$. The algorithm receives a tuple $(f_1, f_2, f_3, f_1^{\alpha}, f_2^{\beta}, f_3^{\tau})$, in which τ is either $\alpha+\beta$ or not, together with the description $(p, \mathbb{G}, \mathbb{G}_T, e, g, h)$ of the bilinear groups. The distinguisher sets up the scheme by choosing $\xi_1, \xi_2, \xi_3, \xi_1', \xi_2', \xi_3', \gamma_0, \gamma_1, \eta_i, \zeta_i, \eta_i', \zeta_i', x_i \leftarrow \mathbb{Z}_p (1 \leq i \leq n)$, $h_0, h_1, h_2 \leftarrow \mathbb{G} \backslash \{1\}$, setting $g_1 = u^{\xi_1} h^{\xi_3}$, $g_2 = v^{\xi_2} h^{\xi_3}$, $vk_0 = h^{\gamma_0}$, $vk_1 = h^{\gamma_2}$ and $A_i = (g h_0^{\zeta} h_1^{u_i} g^{x_i})^{\frac{1}{\gamma_0 + \eta}}$, $B_i = (g h_0^{\zeta_i'} h_1^{u_i} h_2^{t})^{\frac{1}{\gamma_1 + \eta'}}$, and $sk_{OA} = (\xi_1, \xi_2, \xi_3, \xi_1', \xi_2', \xi_3')$, $cert_i = (\langle v_i \rangle, \{A_j\}_{j=0}^{\ell}, g^{x_i})$, $sec_i = x_i$. Queries from the adversary $\mathcal{A}$ to the random oracle H are responded in the ordinary manner, that is, all fresh queries are responded with a random hash value and are recorded together with the hash value, while previously issued queries are responded in the same way as in the previous query. Opening queries are responded as specified in the scheme, that is, the distinguisher first verifies the NIZK proof and if the proof passes the verification, the distinguisher decrypts the linear encryption part $(\psi_1, \ldots, \psi_5)$ using $(\xi_1, \xi_2, \xi_3, \xi_1', \xi_2', \xi_3')$, otherwise return $\perp$. When the adversary requests a challenge regarding (i_0, i_1, M), the distinguisher proceeds as follows: To compute the challenge $(\psi_1^{*}, \ldots, \psi_5^{*}, s_{\alpha}^{*}, \ldots, s_{\beta\eta'}^{*})$, the distinguisher flips a bit b, and sets $\psi_1^{*} = f_1^{\alpha}$, $\psi_2^{*} = f_2^{\beta}$, $\psi_3^{*} = f_3^{\tau}$, $\psi_4^{*} = (\psi_1^{*})^{\xi_1} (\psi_2^{*})^{\xi_2} (\psi_3^{*})^{\xi_3} A_{i_b}$, and $\psi_5^{*} = (\psi_1^{*})^{\xi_1'} (\psi_2^{*})^{\xi_2'} (\psi_3^{*})^{\xi_3'} B_{i_b}$. The zero-knowledge proof $(c^{*}, s_{\alpha}^{*}, \ldots, s_{\beta\eta'}^{*})$ is computed with the simulation algorithm as in Game 1. The distinguisher sends the challenge computed as above to the adversary. After receiving the challenge, the adversary further makes queries to the random oracle and the opening oracle, which are responded as before by the distinguisher. Finally, the adversary outputs the guess b'. The distinguisher outputs 1 if $b = b'$, outputs 0 otherwise.

Observe that when the distinguisher receives a random tuple ($\tau \neq \alpha + \beta$), the adversary's view is equivalent to that of Game 2. In contrast, when the distinguisher receives a linear tuple, we can see that the view is identical to that of Game 1, as the equation $(\psi_1^{*})^{\xi_1} (\psi_2^{*})^{\xi_2} (\psi_3^{*})^{\xi_3} = g_1^{\alpha} g_2^{\beta}$ and $(\psi_1^{*})^{\xi_1'} (\psi_2^{*})^{\xi_2'} (\psi_3^{*})^{\xi_3'} = g_1'^{\alpha} g_2'^{\beta}$ holds. Finally, the lemma follows from the inequality $\Big|\Pr[S_1] - \Pr[S_2]\Big| = \Big|\Pr[\mathcal{B}(f_1, f_2, f_3, f_1^{\alpha}, f_2^{\beta}, f_3^{\tau}) \mid \tau = \alpha + \beta] - \Pr[\mathcal{B}(f_1, f_2, f_3, f_1^{\alpha}, f_2^{\beta}, f_3^{\tau}) \mid \tau \neq \alpha + \beta]\Big| = \mathrm{Adv}_{\mathcal{B}}^{\mathrm{DLIN}}(\lambda)$.

□

Lemma 4.3 $|\Pr[S_2] - \Pr[S_3]| \leq 1/p^2$.

Proof. Since Game 3 differs from Game 2 only when a queried signature, when verified, produces the same $(R_{\alpha}, \ldots, R_{\beta\eta'})$ as the $(R_{\alpha}^{*}, \ldots, R_{\beta\eta'}^{*})$ used in the challenge phase, we examine the mapping $\phi : (R_{\alpha}, R_{\beta}, R_{\alpha+\beta}, R_A, R_B, R_{\alpha\eta}, R_{\beta\eta}, R_{\alpha\eta'}, R_{\beta\eta'}) \rightarrow (s_{\alpha}, R_{\beta}, R_{\alpha+\beta}, R_A, R_B, R_{\alpha\eta}, R_{\beta\eta}, R_{\alpha\eta'}, R_{\beta\eta'})$, implicitly defined by the verification algorithm (notice that the mapping ϕ implicitly depends on the group public key and the signature to be verified), and argue that it is injective with overwhelming probability.

Since the mapping ϕ is a linear function, by calculating the determinant of a matrix we can see that ϕ is injective if and only if $\frac{1}{x_{i^*}+\gamma_0}+(\tau-\beta-\alpha)\xi_3 \log_g h \neq 0$ and $\frac{1}{x_{i^*}+\gamma_1}+(\tau-\beta-\alpha)\xi'_3 \log_g h \neq 0$. Since the adversary can issue queries satisfying the condition $(\psi_1, \dots, \psi_5) = (\psi^*_1, \dots, \psi^*_5)$ and $(R'_\alpha, \dots, R'_{\beta\eta'}) = (R^*_\alpha, \dots, R^*_{\beta\eta'})$ only when the mapping ϕ is injective, the difference $|\Pr[S_2] - \Pr[S_3]|$ is bounded by the probability that the above equation holds. Actually the probability that the equations hold is $1/p^2$, since the random values that appear in the equation are distributed uniformly over $\mathbb{Z}_p$ and independently. □

Lemma 4.4 $|\Pr[S_3] - \Pr[S_4]$ *is negligible.*

Proof. Game 4 differs from Game 3 when the adversary queries the opening oracle with a signature which is not rejected in Game 3 but is rejected in Game 4. We thus bound the probability that the adversary issues such a query. More precisely, the event we consider is that the adversary issues a signature $\sigma = (\psi_1, \dots, \psi_5, s_\alpha, R_\beta, R_{\alpha+\beta}, R_A, R_B, R_{\alpha\eta}, R_{\beta\eta}, R_{\alpha\eta'}, R_{\beta\eta'})$ such that: it is not rejected by the opening oracle, (ψ_1, ψ_2, ψ_3) does not constitute a linear tuple, and $(\psi_1, \dots, \psi_5, R'_\alpha, \dots, R'_{\beta\eta'}) \neq (\psi^*_1, \dots, \psi^*_5, R^*_\alpha, \dots, R^*_{\beta\eta'})$ in which $(R'_\alpha, \dots, R'_{\beta\eta'})$ is the group elements computed in Verify algorithm and $(R^*_\alpha, \dots, R^*_{\beta\eta'})$ are the group elements used for generating challenge signature. If the adversary issues such a query to the opening oracle, there should be the query $(M, \psi_1, \dots, \psi_5, R_\alpha, \dots, R_{\beta\eta'})$ in H (issued by the adversary explicitly or by the opening oracle for verifying the queried signature) such that (ψ_1, ψ_2, ψ_3) does not constitute a linear tuple, and the hash value $H(M, \psi_1, \dots, \psi_5, R_\alpha, \dots, R_{\beta\eta'})$ coincides with the unique challenge c that is determined from the problem instance $(\psi_1, \dots, \psi_5)$ and the commitment $(R_\alpha, \dots, R_{\beta\eta'})$. Hence for concluding the proof it is sufficient to bound the probability of this event. Noticing that in this case any query $(M, \psi_1, \dots, \psi_5, R_\alpha, \dots, R_{\beta\eta'})$ to H in question is different from $(\psi^*_1, \dots, \psi^*_5, R^*_\alpha, \dots, R^*_{\beta\eta'})$ which is used for backpatching, the output of H is chosen from $\mathbb{Z}_p$ uniformly, and thus the probability that a query to H described as above exists with probability less than $q_H + q_{open}/p$ in which q_H and q_{open} respectively denote the upper bounds of the number of queries issued by the adversary to H and the opening oracle, therefore $q_H + q_{open}/p$ is negligible. □

Lemma 4.5 $\Pr[S_4] = 1/2$.

Proof. Here we prove that in this game the value $(\psi^*_1)^{\xi_1}(\psi^*_2)^{\xi_2}(\psi^*_3)^{\xi_3}$ and $(\psi^*_1)^{\xi'1}(\psi^*_2)^{\xi'2}(\psi^*_3)^{\xi'3}$ are uniformly random even when conditioned on the adversary's view. To this end we examine the distribution of the adversary's view related to the randomness $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$ under the condition where all the other randomness involved in the game are fixed. The adversary obtains information related $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$ from the part of the group public key g_1, g_2, g'_1, g'_2 and the responses from the opening oracle. As for the responses from the opening oracle, any query whose ϕ_1, ψ_2, ψ_3 components does not constitute the linear tuple will be rejected by the opening oracle, thus the adversary gains no information on $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$ from such queries. A query with a linear tuple also gives no information to the adversary. When the adversary issues a signature $(\psi_1, \dots, \psi_5, c, s_\alpha, \dots, s_{\beta\eta'})$, the opening oracle computes group elements $(\psi^*_1)^{\xi_1}(\psi^*_2)^{\xi_2}(\psi^*_3)^{\xi_3}$ and $(\psi^*_1)^{\xi'_1}(\psi^*_2)^{\xi'_2}(\psi^*_3)^{\xi'_3}$ (the rest of the calculation performed by the oracle is done without referring to $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$), which is what the adversary learns from this query. It in fact does not increase the information the adversary knows, since the above equation can be rewritten as $(\psi^*_1)^{\xi_1}(\psi^*_2)^{\xi_2}(\psi^*_3)^{\xi_3} = (f_1^\alpha)^{\xi_1}(f_2^\beta)^{\xi_2}(f_3^{\alpha+beta})^{\xi_3} = (f_1^{\xi_1} f_3^{\xi_3})^\alpha (f_2^{\xi_2} f_3^{\xi_3})^\beta = g_1^\alpha g_2^\beta$ and $(\psi^*_1)^{\xi'_1}(\psi^*_2)^{\xi'_2}(\psi^*_3)^{\xi'_3} = g_1^\alpha g_2^\beta$ can be similarly obtained, when we write $\psi_1 = f_1^\alpha$, $\psi_2 = f_2^\beta$, and $\psi_3 = f_3^{\alpha+\beta}$. The right-hand side of the equation shows that the response of the opening oracle gives no information to the adversary, since all the values that appears in the right-hand side are already known to the adversary.

The above discussion shows that the responses of the opening oracles do not leak any information of $\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3$. Finally we shows that the value $(\psi^*_1)^{\xi_1}(\psi^*_2)^{\xi_2}(\psi^*_3)^{\xi_3}$ and $(\psi^*_1)^{\xi'_1}(\psi^*_2)^{\xi'_2}(\psi^*_3)^{\xi'_3}$ are uniformly distributed conditioned on the group public key g_1, g_2, g'_1, g'_2. This can be done by considering the following equation

$$\begin{pmatrix} \log_g g_1 \\ \log_g g_2 \\ \log_g (\psi^*_1)^{\xi_1}(\psi^*_2)^{\xi_2}(\psi^*_3)^{\xi_3} \end{pmatrix} = \begin{pmatrix} t_1 & 0 & t_3 \\ 0 & t_2 & t_3 \\ t_1\alpha & t_2\beta & t_3\tau \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix},$$

where $t_1 = \log_g f_1$, $t_2 = \log_g f_2$, and $t_3 = \log_g f_3$. Since the matrix in the right-hand side has the determinant $t_1 t_2 t_3(\tau - \alpha - \beta) \neq 0$, the value $(\psi_1^*)^{\xi_1}(\psi_2^*)^{\xi_2}(\psi_3^*)^{\xi_3}$ is distributed uniformly and independently of g_1 and g_2. This shows that the challenge signature is independent of A_{i_b} and hence of the challenge bit b. □

From the above, the proof of Theorem 4.1 is completed. □

Theorem 4.2 *The proposed R-GS scheme has non-frameability in the random oracle model under the DL assumption, where H is modeled as a random oracle.*

The discrete logarithm x of X in the membership certificate is the secret information that only the signer knows, and the signer issues the NIZK proof for the knowledge of x in the signing. Therefore, it seems that the signature cannot be forged without x. In the simulation of the adversary against the DL problem, the extractor of x can be construct by rewinding the adversary against non-frameability (this proof is based on the forking lemma [56]). Therefore, the non-frameability of this scheme is reduced to the DL problem.

Proof. The adversary $\mathcal{A}$ comes up with a forgery (M^*, σ^*) that opens to some honest user $i \in U^b$ and that did not issue a signature.

Given a problem instance $(g, y = g^x, p, \mathbb{G}_1)$, the simulator $\mathcal{B}$ generates $(f_1, f_2, f_3, h_0, h_1, h_2) \leftarrow \mathbb{G}_1 \setminus \{1\}$, $h \leftarrow \mathbb{G}_2$, $\gamma_0, \gamma_1, \xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3 \leftarrow \mathbb{Z}_p$, then compute $(sk_0, vk_0) = (\gamma_0, h^{\gamma_0})$, $(sk_1, vk_1) = (\gamma_1, h^{\gamma_1})$, $g_1 = f_1^{\xi_1} f_3^{\xi_3}$, $g_2 = f_2^{\xi_2} f_3^{\xi_3}$, $g'_1 = f_1^{\xi'_1} f_3^{\xi'_3}$, $g'_2 = f_2^{\xi'_2} f_3^{\xi'_3}$, set $sk_{OA} = (\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3)$, $sk_{GM} = (sk_0, sk_1)$, $gpk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, f_1, f_2, f_3, g_1, g_2, g'_1, g'_2, h_0, h_1, h_2, h, vk_0, vk_1, H)$.

At the beginning of the game, $\mathcal{B}$ picks a random index $j^* \leftarrow \{1, \dots, q_{join}\}$ of the $O_{b\text{-}join}$ query. Then, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

- O_{gpk}, O_{GM}, O_{OA} query: $\mathcal{B}$ returns gpk, sk_{GM} and sk_{OA} as described above, respectively.
- $O_{b\text{-}join}$ query: When $\mathcal{A}$ (as group manager) requests to run Join protocol for a new honest user i in the group, $\mathcal{B}$ executes J_{user}. Depending on the index of $O_{b\text{-}join}$ queries, $\mathcal{B}$ behaves as follows:
  - If $i \neq i^*$, $\mathcal{B}$ follows J_{user} exactly.
  - If $i = i^*$, $\mathcal{B}$ sends the value $y = g^x$ as X. In subsequent steps of the Join protocol, $\mathcal{B}$ proceeds as the real J_{user}. When Join terminates, $\mathcal{B}$ obtain a membership certificate $cert_i = (\langle v_i \rangle, \{A_j\}_{j=0}^{\ell}, y)$.
- O_{revoke} query: It can be treated as the real game, since $\mathcal{B}$ has sk_{GM}.
- O_{sig} query: When $\mathcal{A}$ asks a signature for a message M of the user $i \in U^b$, $\mathcal{B}$ treats as follows:
  - If $i \neq j^*$, $\mathcal{B}$ can simulate the signing algorithm as the real game.
  - If $i = j^*$, $\mathcal{B}$ generates the signature using $cert_{j^*}$, issued by the j^*th query of $O_{b\text{-}join}$.

Finally, $\mathcal{A}$ outputs a signature $\sigma^* = (\psi_1^*, \dots, \psi_5^*, c^*, s_\alpha^*, \dots, s_{\beta\eta'}^*)$, for some message M^*, that opens to some user $i^* \in U^b$ who did not sign M^*. Then, $\mathcal{B}$ computes $A_{i^*} = \psi_4 / \psi_1^{\xi_1} \psi_2^{\xi_2} \psi_3^{\xi_3}$. If there exists the transcript $(\langle v_i \rangle, \{A_j\}_{j=0}^{\ell}, X)$ such that $X = y = g^x$, we apply forking lemma [56] and obtain the discrete logarithm x of $y = g^x$, then output x. Otherwise, $\mathcal{B}$ outputs $\perp$ and halts.

For proving the theorem, we use the forking lemma [56].

Lemma 4.6 (Forking Lemma) *Fix an integer $q_H \geq 1$. Let $\mathcal{A}$ be a randomized algorithm that on input $x, h_1, \dots, h_{q_H}$, where x is a random source for running $\mathcal{A}$, $h_1, \dots, h_{q_H}$ are the responses from the random oracle. The acceptance probability of $\mathcal{A}$, denoted acc(k) is defined as* $acc(\lambda) = \Pr[i \geq 1 \mid vk \leftarrow \mathrm{Gen}(1^\lambda); x \leftarrow \mathcal{R}; h_1, \dots, h_{q_H} \leftarrow \mathbb{Z}_p; i = \mathcal{A}(vk, x)^{(h_1, \dots, h_{q_H})}]$. *The forking algorithm $\mathcal{B}$ corresponds with $\mathcal{A}$ is a randomized algorithm proceed as follows: (1) $x \leftarrow \mathcal{R}$, (2) $h_1, \dots, h_{q_H}, h'_1, \dots, h'_{q_H} \leftarrow \mathbb{Z}_p$, (2)*

$i \leftarrow \mathcal{A}(vk, x)^{(h_1,\ldots,h_{q_H})}$, (4) $i' \leftarrow \mathcal{A}(vk, x)^{(h_1,\ldots,h_{i-1},h'_i,\ldots,h'_{q_H})}$, (5) Outputs 1 if $(i = i') \wedge (i \neq 0) \wedge (h_i \neq h'_{i'})$, otherwise outputs 0.

Let $frk(k)$ be the probability that $\mathcal{B}$ outputs 1. Then, the following equation $frk(k) \geq acc(k) \cdot \left(\frac{acc(k)}{q_H} - \frac{1}{p}\right)$ holds.

Next, we prove the following lemma by using the forking lemma.

Lemma 4.7

$$\mathrm{Adv}_{\mathcal{A}}^{frame}(\lambda) \leq \left(q_{b\text{-}join} \cdot q_{sig} \cdot \mathrm{Adv}_{\mathcal{B}}^{\mathrm{DL}}(\lambda) + \frac{(1+q_{sig})}{p}\right)^{1/2}$$

Proof.

$$frk(\lambda) \geq \mathrm{Adv}_{\mathcal{A}}^{\mathrm{frame}}(\lambda) \cdot \left(\frac{\mathrm{Adv}_{\mathcal{A}}^{\mathrm{frame}}(\lambda) - 1/p}{q_{sig}} - \frac{1}{p}\right) > \frac{(\mathrm{Adv}_{\mathcal{A}}^{\mathrm{frame}}(\lambda))^2}{q_{sig}} - \frac{1 + 1/q_{sig}}{p}.$$

Moreover, if $\mathcal{A}$ outputs forged signature, $\mathcal{B}$ outputs the discrete logarithm with probability $1/q_{b\text{-}join}$. That is, $\frac{(\mathrm{Adv}_{\mathcal{A}}^{\mathrm{frame}}(\lambda))^2}{q_{sig}} - \frac{1+1/q_{sig}}{p} \leq q_{b\text{-}join} \cdot \mathrm{Adv}_{\mathcal{B}}^{\mathrm{DL}}(\lambda)$ holds. □

From the forking lemma, it is shown that $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{frame}}(\lambda)$ is negligible if $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{DL}}(\lambda)$ is negligible.

□

Theorem 4.3 *The proposed R-GS scheme has misidentification resistance in the random oracle model under the q-SDH assumption and knowledge of secret key (KOSK) assumption [57], where H is modeled as a random oracle.*

The misidentification attack means a forgery of the BBS+ signature as membership certificate. Hence, the security against misidentification attacks can be reduced to the unforgeability of the BBS+ signature scheme, and it is proved in [6]. We consider two types of forgers: (1) forgery of the certificate for belonging to the group, and (2) forgery of the certificate of the non-revoked users. Since breaking the unforgeability of the BBS+ signature scheme allows us to construct an algorithm that breaks the q-SDH assumption, the theorem holds.

In the actual Join algorithm, a user sends $X = h_2^x$ to the group manager, and the group manager, who has the signing key of the BBS+ signature scheme, can sign x without knowing x itself, and can make a certificate A. Whereas, in the security proof, the simulator needs to send a signed message x in order to access the signing oracle of the underlying BBS+ signature scheme. However, since an adversary sends not x but X to the simulator, we need to consider how to obtain the corresponding x in the security proof. To circumvent this obstacle, one solution is to add the proof of knowledge of the secret key in the beginning of the Join algorithm, and extract x by rewinding the adversary. However, we need to rewind the adversary a number of queried times. This requires much loose reduction cost, and it seems difficult to estimate the actual success probability of the extraction. Therefore, we introduce the knowledge of secret key (KOSK) assumption [57] where the adversary is required to reveal the secret key of the honest users, which is joined by $O_{a\text{-}join}$ queries. In addition, we assume that $O_{a\text{-}join}$ queries are not executed concurrently.

In type (1) forgery, we simulate a group manager who implements the join protocol. In this simulation, he/she gets $X = h_2^x$ from a user and sends A_j to the user. The group manager needs $x = \log_{h_2} X$ that is sent to the signature oracle in order to get signature A_j. On the other hand, in type (2), we simulate a group manager who implements the Revoke algorithm. In this simulation, he/she can make B_j that is a part of revocation list RL_t without proof of knowledge.

Proof. The adversary $\mathcal{A}$ comes up with a forgery (M^*, σ^*) that doesn't open to non-revoked dishonest user $i \in U^a \setminus R_{t^*}$ and that did not issue a signature. We will argue that the simulator $\mathcal{B}$ that breaks the BBS+ signature (which is secure under q-SDH assumption) can be constructed from the adversary $\mathcal{A}$ that breaks the misidentification resistance of the proposed scheme.

At the beginning of the game, $\mathcal{B}$ picks a random index $j^* \leftarrow \{1, \ldots, q_{join}\}$ of the $O_{a\text{-}join}$ query. We consider two types of adversary. Given a problem instance $(g, w = h^{\gamma}, p, \mathbb{G}_1)$ (public key of the BBS+ signature), the simulator $\mathcal{B}$ generates $(f_1, f_2, f_3, h_0, h_1, h_2) \leftarrow \mathbb{G}_1 \setminus \{1\}$, $h \leftarrow \mathbb{G}_2$, $\gamma_0, \gamma_1, \xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3 \leftarrow \mathbb{Z}_p$. If we consider type (1) forger, set $vk_0 = w$, and compute $(sk_1, vk_1) = (\gamma_1, h^{\gamma_1})$. Otherwise (i.e., consider type (2) forger), set $vk_1 = w$, and compute $(sk_0, vk_0) = (\gamma_0, h^{\gamma_0})$. Then, compute $g_1 = f_1^{\xi_1} f_3^{\xi_3}$, $g_2 = f_2^{\xi_2} f_3^{\xi_3}$, $g'_1 = f_1^{\xi'_1} f_3^{\xi'_3}$, $g'_2 = f_2^{\xi'_2} f_3^{\xi'_3}$, set $sk_{OA} = (\xi_1, \xi_2, \xi_3, \xi'_1, \xi'_2, \xi'_3)$, $sk_{GM} = (sk_0, sk_1)$, $gpk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, f_1, f_2, f_3, g_1, g_2, g'_1, g'_2, h_0, h_1, h_2, h, vk_0, vk_1, H)$.

- Type (1) forger: The adversary $\mathcal{A}$ forges a certificate A_{j^*} for that belonging to the group, which corresponds with $O_{a\text{-}join}$ queries.
- Type (2) forger: The adversary $\mathcal{A}$ forges a certificate B_{j^*,t^*} for that the non-revoked member, which corresponds with O_{revoke} queries.

$\mathcal{B}$ interacts $\mathcal{A}$ as follows:

- O_{gpk} query: $\mathcal{B}$ returns gpk as described above to $\mathcal{A}$.
- $O_{a\text{-}join}$ query: When $\mathcal{A}$ requests to run Join protocol for a (corrupted) user i in the group, $\mathcal{B}$ executes J_{GM}. $\mathcal{B}$ behaves as follows:
  - Type (1) forger: When $\mathcal{A}$ sends a group elements X, $\mathcal{A}$ also sends $x = \log_{h_2} X$ (by the KOSK assumption). Then $\mathcal{B}$ asks the signing oracle of the BBS+ signature to generate a (part of) certificate $A_j = (g h_0^{\zeta_j} h_1^{u_j} h_2^{x})^{\frac{1}{\gamma+\eta_j}}$. In the subsequent step of the Join protocol, $\mathcal{B}$ proceeds as the real J_{GM}.
  - Type (2) forger: It can be treated as in real game.
- O_{revoke} query: When $\mathcal{A}$ asks to revoke the user i, $\mathcal{B}$ behaves as follows.
  - Type (1) forger: $\mathcal{B}$ generates new certificates for non-revoked users $\{\Theta = (B_{i,t}, \eta'_i, \zeta'_i)\}_{i=0}^{\mathtt{num}}$. It can be treated as in real game, since $\mathcal{B}$ has sk_{GM}.
  - Type (2) forger: $\mathcal{B}$ generates the BBS+ signature $\{B_{i,t} = (g h_0^{\zeta'_i} h_1^{u_i} h_2^{t})^{\frac{1}{\eta'_i+\gamma}}\}$ by asking the signing oracle of BBS+ signatures. In the subsequent steps, $\mathcal{B}$ proceeds as the real game. new certificates for non-revoked users $\{\Theta = (B_{i,t}, \eta'_i, \zeta'_i)\}_{i=0}^{\mathtt{num}}$.

Finally, $\mathcal{A}$ outputs a signature $\sigma^* = (\psi_1^*, \ldots, \psi_5^*, c^*, s_{\alpha}^*, \ldots, s_{\beta\eta'}^*)$, for some message M^*, that opens to some user $i^* \in U^a \setminus R_{t^*}$ who did not sign M^*. Then, $\mathcal{B}$ decrypts ψ_4 to get A_{i^*} (if type (1) forger) or decrypts ψ_5 to get B_{i^*,t^*} (if type (2) forger). $\mathcal{B}$ outputs the decrypted certificates as the forged signature. Let the events that $\mathcal{A}$ of Type (1) and (2) succeed to forge the BBS+ signature be F_1 and F_2, respectively. From the above game, B can forge the BBS+ signature if $\mathcal{A}$ wins the misidentification game. Therefore, the following equation $\mathrm{Adv}_{\mathcal{A}}^{misid}(\lambda) \leq \Pr[F_1] + \Pr[F_2]$ holds. The probability $\Pr[F_1]$ and $\Pr[F_2]$ are negligible if the q-SDH assumption holds since the BBS+ signature is unforgeable under q-SDH assumption. Therefore, it is shown that $\mathrm{Adv}_{\mathcal{A}}^{misid}(\lambda)$ is negligible if the q-SDH assumption holds. □

4.4 Discussion on the construction of SD-based R-GS scheme

In this section, again we consider to construct a SD-based scheme in the random oracle model in an efficient way. By using the SD method, all (non-revoked) users are partitioned as $S_1, \ldots, S_{\mathsf{num}}$ (as in the CS method) where $\mathsf{num} = O(r)$. Each set S_j is described as $S_j := S_{k_j, k'_j}$ where k_j and k'_j are node of the tree of level ϕ_j and ψ_j, respectively. If a signer who has certificates of $u_0, u_1, ..., u_\ell$ is not revoked, that the condition (1) $k_j = u_{\phi_j}$ (k_j is an ancestor of the leaf node that the signer is assigned) and (2) $k'_j \neq u_{\psi_j}$ (k'_j is not an ancestor of the leaf node) must hold. The first "equality" condition can be proved as in our

CS-method based scheme. However, it is not trivial to prove the second "inequality" condition where signed messages of two BBS+ signatures are different without showing messages themselves. In [44], this inequality relation is proved by using the Boneh–Boyen signature with the form $g^{1/(k'_j - u_{\psi_j})}$ and Groth–Sahai proofs for the verification pairing equation of the signature. As mentioned in the paper, the languages of the Groth–Sahai proof and those of the Fiat–Shamir proof are completely different. This is the first obstacle.

Even if we can solve this problem, the next problem is efficiency (signature size). That is, we need to prove (at least) one more relation compared to the CS method based scheme. More precisely, in our scheme two BBS group signature schemes are run for A and B_t, respectively, and we need to (at least) run one more BBS group signature if the inequality relation needs to be proved additionally. This means the signature size is at least 1.5 times longer than that of the proposed scheme. This is the second obstacle.

From the above discussions, it seems not trivial to efficently construct a SD-based scheme in the random oracle model. We leave it as a future work of this paper.

5 Conclusion

In this paper, we proposed a scalable R-GS group signature scheme with compact signature size. In order to efficiently implement the scheme, we used the parallel BBS group signature technique where two BBS group signature schemes are simultaneously run but a part of random values are commonly used. By using this technique, we do not have to apply broadcast encryption which was used in the LPY schemes. Since random oracles break underlying algebraic structures, it seems not trivial to achieve constant certificate size [44] or constant revocation list [4, 52] without detracting the current efficiency. We leave it as an interesting future work of this paper.

Acknowledgement: We thank the members of Shin-Akarui-Angou-Benkyou-Kai for their helpful comments.

References

[1] ISO/IEC 18033-2:2006 information technology – security techniques – encryption algorithms – part 2: Asymmetric ciphers.

[2] ISO/IEC 20008-2:2013 information technology – security techniques – anonymous digital signatures – part 2: Mechanisms using a group public key.

[3] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO*, pages 255–270. 2000.

[4] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. A revocable group signature scheme from identity-based revocation techniques: Achieving constant-size revocation list. In *ACNS*, pages 419–437, 2014.

[5] N. Attrapadung, B. Libert, and E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *PKC*, pages 90–108, 2011.

[6] M. H. Au, W. Susilo, Y. Mu, and S. S. M. Chow. Constant-size dynamic k-times anonymous authentication. *IEEE Systems Journal*, 7(2):249–261, 2013.

[7] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *EUROCRYPT*, pages 1–16, 2014.

[8] N. Begum, T. Nakanishi, and N. Funabiki. Efficient proofs for CNF formulas on attributes in pairing-based anonymous credential system. In *ICISC*, pages 495–509, 2012.

[9] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.

[10] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, pages 136–153, 2005.

[11] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get Shorty via Group Signatures without Encryption. In *SCN*, pages 381–398. 2010.

[12] D. Boneh and X. Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. *Journal of Cryptology*, 21(2):149–177, 2008.

[13] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[14] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *CRYPTO*, pages 227–242. 2004.

[15] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[16] D. Boneh and H. Shacham. Group Signatures with Verifier-Local Revocation. In *ACM CCS 2004*, pages 168–177. ACM, New York, 2004.

[17] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of fully dynamic group signatures. Cryptology ePrint Archive, Report 2016/368, 2016. `http://eprint.iacr.org/`.

[18] E. F. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. In *Submission to the Trusted Computing Group*, 2003.

[19] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, pages 56–72. 2004.

[20] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN*, pages 57–75, 2012.

[21] D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT*, pages 257–265. 1991.

[22] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee. Shorter IBE and Signatures via Asymmetric Pairings. In *Pairing*, pages 122–140, 2012.

[23] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[24] C. Delerablée and D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *VIETCRYPT*, pages 193–210. 2006.

[25] D. Derler and D. Slamanig. Fully-anonymous short dynamic group signatures without encryption. Cryptology ePrint Archive, Report 2016/154, 2016. `http://eprint.iacr.org/`.

[26] Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In *Digital Rights Management Workshop*, pages 61–80, 2002.

[27] C.-I. Fan, R.-H. Hsu, and M. Manulis. Group Signature with Constant Revocation Costs for Signers and Verifiers. In *CANS*, pages 214–233, 2011.

[28] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[29] J. Furukawa and H. Imai. An Efficient Group Signature Scheme from Bilinear Maps. *IEICE Transactions*, 89-A(5):1328–1338, 2006.

[30] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.

[31] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT*, pages 395–412, 2010.

[32] R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *CRYPTO*, pages 126–145, 2014.

[33] J. Groth. Fully Anonymous Group Signatures Without Random Oracles. In *ASIACRYPT*, pages 164–180, 2007.

[34] J. Groth and A. Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. In *EUROCRYPT*, pages 415–432, 2008.

[35] J. Y. Hwang, S. Lee, B. ho Chung, H. S. Cho, and DaeHunNyang. Short group signatures with controllable linkability. *LightSec*, pages 44–52, 2011.

[36] A. Kiayias and M. Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. *IACR Cryptology ePrint Archive*, 2004:76, 2004.

[37] A. Kiayias and M. Yung. Group Signatures with Efficient Concurrent Join. In *EUROCRYPT*, pages 198–214. 2005.

[38] A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *IJSN*, 1(1/2):24–45, 2006.

[39] V. Kumar, H. Li, J. J. Park, K. Bian, and Y. Yang. Group signatures with probabilistic revocation: A computationally-scalable approach for providing privacy-preserving authentication. In *ACM CCS*, pages 1334–1345, 2015.

[40] K. Kurosawa. Multi-recipient Public-Key Encryption with Shortened Ciphertext. In *PKC*, pages 48–63, 2002.

[41] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT*, pages 41–61, 2013.

[42] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC*, pages 345–361, 2014.

[43] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. Cryptology ePrint Archive, Report 2016/101, 2016. `http://eprint.iacr.org/`.

[44] B. Libert, T. Peters, and M. Yung. Group Signatures with Almost-for-Free Revocation. In *CRYPTO 2012*, pages 571–589, 2012.

[45] B. Libert, T. Peters, and M. Yung. Scalable Group Signatures with Revocation. In *EUROCRYPT*, pages 609–627, 2012.

[46] B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, pages 296–316, 2015.

[47] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *CANS*, pages 498–517, 2009.

[48] B. Libert and M. Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In *TCC*, pages 499–517, 2010.

[49] S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC*, pages 427–449, 2015.

[50] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. *IEICE Transactions*, 93-A(1):50–62, 2010.

[51] T. Nakanishi and N. Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. *IEICE Transactions*, 90-A(1):65–74, 2007.

[52] T. Nakanishi and N. Funabiki. Revocable group signatures with compact revocation list using accumulators. In *ICISC*, pages 435–451, 2013.

[53] D. Naor, M. Naor, and J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. *Electronic Colloquium on Computational Complexity (ECCC)*, (043), 2002.

[54] P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC*, pages 401–426, 2015.

[55] D. Pointcheval and O. Sanders. Short randomizable signatures. In *CT-RSA*, pages 111–126, 2016.

[56] D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *EUROCRYPT*, LNCS, pages 387–398. 1996.

[57] T. Ristenpart and S. Yilek. The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks. In *EUROCRYPT*, pages 228–245, 2007.

[58] Y. Sakai, J. C. N. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *PKC*, pages 715–732, 2012.

[59] H. Shacham. A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/`.