Scalable Group Signatures with Revocation

Benoît Libert[1] ⋆, Thomas Peters[1] ⋆⋆, and Moti Yung[2]

[1]Université catholique de Louvain, ICTEAM Institute (Belgium)
[2] Google Inc. and Columbia University (USA)

Abstract. Group signatures are a central cryptographic primitive, simultaneously supporting accountability and anonymity. They allow users to anonymously sign messages on behalf of a group they are members of. The recent years saw the appearance of several constructions with security proofs in the standard model (*i.e.*, without appealing to the random oracle heuristic). For a digital signature scheme to be adopted, an efficient revocation scheme (as in regular PKI) is absolutely necessary. Despite over a decade of extensive research, membership revocation remains a non-trivial problem in group signatures: all existing solutions are not truly scalable due to either high overhead (e.g., large group public key size), or limiting operational requirement (the need for all users to follow the system's entire history). In the standard model, the situation is even worse as many existing solutions are not readily adaptable. To fill this gap and tackle this challenge, we describe a new revocation approach based, perhaps somewhat unexpectedly, on the Naor-Naor-Lotspiech framework which was introduced for a different problem (namely, that of broadcast encryption). Our mechanism yields efficient and scalable revocable group signatures in the standard model. In particular, the size of signatures and the verification cost are independent of the number of revocations and the maximal cardinality N of the group while other complexities are at most polylogarithmic in N. Moreover, the schemes are history-independent: unrevoked group members do not have to update their keys when a revocation occurs.

Keywords. Group signatures, revocation, standard model, efficiency.

1 Introduction

As suggested by Chaum and van Heyst in 1991 [32], *group signatures* allow members of a group to anonymously sign messages on behalf of a population group members managed by a group authority. Using some trapdoor information, a tracing authority must be able to "open" signatures and identify the signer. A complex problem in group signatures is the revocation of members whose signing capability should be disabled (either because they misbehaved or they intentionally leave the group).

1.1 Related Work

Group signatures without revocation. The first provably coalition-resistant scalable group signature was described by Ateniese, Camenisch, Joye and Tsudik in 2000 [7]. At that time, the security of group signatures was not totally understood and proper security definitions were given later on by Bellare, Micciancio and Warinschi [9] (BMW) whose model captures all the requirements of group signatures in three properties. In (a relaxation of) this model, Boneh, Boyen and Shacham [16] obtained a construction in the random oracle model [10] with signatures shorter than 200 bytes [13].

In the BMW model, the population of users is frozen after the setup phase beyond which no new member can be added. Dynamic group signatures were independently formalized by Kiayias and Yung [45] and Bellare-Shi-Zhang [11]. In these models, pairing-based schemes with relatively short signatures were put forth in [54, 33]. Ateniese *et al.* [6] also gave a construction without random oracles using interactive assumptions. In the BMW model [9], Boyen and Waters independently came up with a different standard model proposal [19] using more classical assumptions and they subsequently refined their scheme [21] to

⋆ This author was supported by the Belgian Fund for Scientific Research (F.R.S.-F.N.R.S.) via a "Chargé de recherches" fellowship.
⋆⋆ Supported by the IUAP B-Crypt Project and the Walloon Region Camus Project.

obtain constant-size signatures. In the dynamic model [11], Groth [38] described a system with constant-size signatures without random oracles but this scheme was rather a feasibility result than an efficient construction. Later on, Groth gave [39] a fairly efficient realization – with signatures consisting of about 50 group elements – in the standard model with the strongest anonymity level.

REVOCATION. In group signatures, membership revocation has received much attention in the last decade [22, 8, 29, 18] since revocation is central to digital signature schemes. One simple solution is to generate a new group public key and deliver a new signing key to each unrevoked member. However, in large groups, it may be inconvenient to change the public key and send a new secret to signers after they joined the group. An alternative approach taken by Bresson and Stern [22] is to have the signer prove that his membership certificate does not appear in a public list or revoked certificates. Unfortunately, the signer's workload and the size of signatures grow with the number of expelled users.

Song [55] presented an approach handling revocation in forward-secure group signatures. However, verification takes linear time in the number of excluded users.

Using accumulators[1] [12], Camenisch and Lysyanskaya [29] proposed a method (notably followed by [60, 27]) to revoke users in the ACJT group signature [7] while keeping $O(1)$ costs for signing and verifying. While elegant, this approach is history-dependent and requires users to keep track of all changes in the population of the group: at each modification of the accumulator value, unrevoked users need to update their membership certificates before signing new messages, which may require $O(r)$ exponentiations – if r is the number of revoked users – in the worst case.

Brickell [23] suggested the notion of *verifier-local revocation* group signatures, which was formalized by Boneh and Shacham [18] and further studied in [50, 61, 48]. In their systems, revocation messages are only sent to verifiers (making the signing algorithm independent of the number of revocations). The group manager maintains a revocation list (RL) which is used by verifiers to make sure that signatures were not generated by a revoked member. The RL contains a token for each revoked user and the verification algorithm has to verify signatures w.r.t. each token (a similar revocation mechanism is used in [24]). As a result, the verification cost is inevitably linear in the number of expelled users.

More recently, Nakanishi, Fuji, Hira and Funabiki [49] described a construction with constant complexities for signing/verifying and where group members never have to update their credentials. On the other hand, their proposal has the disadvantage of linear-size group public keys (in the maximal number N of users), although a tweak allows reducing the size to $O(N^{1/2})$.

In the context of anonymous credentials, Tsang *et al.* [58, 59] showed how to blacklist users without compromising their anonymity or involving a trusted third party. Their protocols either have linear proving complexity in the number of revocations or rely on accumulators (which may be problematic for our purposes). Camenisch, Kohlweiss and Soriente [28] suggested to handle revocations by periodically updating users credentials in which a specific attribute indicates a validity period. While useful in certain applications of anonymous credentials, in group signatures, their technique would place quite a burden on the group manager who would have to generate updates for each unrevoked individual credential.

1.2 Our Contribution

For the time being and despite over a decade of research efforts, group signatures in the standard model have no revocation mechanism allowing for scalable (*i.e.*, constant or polylogarithmic) verification time without dramatically degrading the efficiency in other metrics and without being history-dependent. In pairing-based group signatures, accumulator-based approaches are unlikely to result in solutions supporting very large groups. The reason is that, in known pairing-based accumulators [53, 27], public keys have linear size in the maximal number of accumulated values (unless one sacrifices the constant size of proofs of

[1] An accumulator allows hashing a set of values into a short string of constant size while allowing to efficiently prove that a specific value was accumulated.

non-membership as in [5]), which would result in linear-size group public keys in straightforward implementations. Recently [35], Fan *et al.* suggested a different way to use the accumulator of [27] and announced constant-size group public keys but their scheme still requires the group manager to publicize $O(N)$ values at each revocation. In a revocation mechanism along the lines of [29], Boneh, Boyen and Shacham [16] managed to avoid linear dependencies. However, their technique seems hard to combine[2] with Groth-Sahai proofs [40] so as to work in the standard model, which is our goal. In addition, we would like to save unrevoked users from having to update their keys after each revocation. To this end, it seems possible to adapt the approach of Nakanishi *et al.* [49] in the standard model. However, merely replacing sigma-protocols by Groth-Sahai proofs in the scheme of [49] would result in group public keys of size $O(N^{1/2})$ in the best case.

In this paper, we describe a novel and scalable revocation technique that interacts nicely with Groth-Sahai proofs and gives constructions in the standard model with $O(1)$ verification cost and at most polylogarithmic complexity in other metrics. Our approach bears similarities with the one of Nakanishi *et al.* [49] in that it does not require users to update their membership certificates at any time but, unlike [49], our group public key size is either $O(\log N)$ or constant. Like the scheme of [49], our main system uses revocation lists (RLs) of size $O(r)$ – which is in line with certificate revocation lists of standard PKIs – and we emphasize that these are *not* part of the group public key: verifiers only need to know the number of the latest revocation epoch and they do not have to read RLs entirely.

To obtain our constructions, we turn to the area of broadcast encryption and build on the Subset Cover framework of Naor, Naor and Lotspiech [51] (NNL). In a nutshell, the idea is to use the NNL ciphertext as a revocation list and have non-revoked signers prove their ability to decrypt in order to convince verifiers that they are not revoked. In its public-key variant, due to Dodis and Fazio [34], the Subset Cover framework relies on hierarchical identity-based encryption (HIBE) [44, 37] and each NNL ciphertext consists of several HIBE encryptions. To anonymously sign a message, we let group members commit to the specific HIBE ciphertext that they can decrypt (which gives constant-size signatures since only one ciphertext is committed to), and provide a non-interactive proof that: (i) they hold a private key which decrypts the committed HIBE ciphertext. (ii) The latter belongs to the revocation list.

By applying this approach to the Subset Difference (SD) method [51], we obtain a scheme with $O(1)$-size signatures, $O(\log N)$-size group public keys, membership certificates of size $O(\log^3 N)$ and revocation lists of size $O(r)$. The Layered Subset Difference method [41] can be used in the same way to obtain membership certificates of size $O(\log^{2.5} N)$ while retaining the same efficiency elsewhere. Using the Complete Subtree method, we also obtain a tradeoff with $O(r \cdot \log N)$ revocation lists, log-size membership certificates and constant-size group public keys (comparisons among schemes are given in Section 4).

A natural question is whether our SD-based revocable group signatures can generically use any HIBE scheme. The answer is negative as the Boneh-Boyen-Goh (BBG) construction [15] is currently the only suitable candidate. For anonymity reasons, ciphertexts should be of constant size and our security proof requires the HIBE system to satisfy a new and non-standard security property which is met by [15]. As we will see, the proof can hardly rely on the standard security notion for HIBE schemes [37].

We note that the new revocation mechanism can find applications in contexts other than group signatures. For example, it seems that it can be used in the oblivious transfer with access control protocol of [26], which also uses the technique of Nakanishi *et al.* [49] to revoke credentials.

[2] In the scheme of [16], signing keys consist of pairs $(g^{1/(\omega+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$, where $\omega \in \mathbb{Z}_p$ is the private key of the group manager, and the revocation mechanism relies on the availability of the exponent $s \in \mathbb{Z}_p$. In the standard model, the Groth-Sahai techniques would require to turn the membership certificates into triples $(g^{1/(\omega+s)}, g^s, u^s)$, for some $u \in \mathbb{G}$ (as in [21]), which is no longer compatible with the revocation technique.

2 Background

2.1 Bilinear Maps and Complexity Assumptions

We use bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p where $e(g,h) \neq 1_{\mathbb{G}_T}$ if and only if $g, h \neq 1_{\mathbb{G}}$. In these groups, we rely on hardness assumptions that are all falsifiable [52].

Definition 1 ([16]). *The* **Decision Linear Problem** *(DLIN) in $\mathbb{G}$, is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \xleftarrow{R} \mathbb{Z}_p^*$, $z \xleftarrow{R} \mathbb{Z}_p^*$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher* D.

Definition 2 ([13]). *The* **q-Strong Diffie-Hellman problem** *(q-SDH) in $\mathbb{G}$ is, given $(g, g^a, \ldots, g^{(a^q)})$, for some $g \xleftarrow{R} \mathbb{G}$ and $a \xleftarrow{R} \mathbb{Z}_p$, to find a pair $(g^{1/(a+s)}, s) \in \mathbb{G} \times \mathbb{Z}_p$.*

Finally, we appeal to yet another "q-type" assumption introduced by Abe *et al.* [2].

Definition 3 ([2]). *In a group $\mathbb{G}$, the* **q-Simultaneous Flexible Pairing Problem** *(q-SFP) is, given $\big(g_z,\ h_z,\ g_r,\ h_r,\ a,\ \tilde{a},\ b,\ \tilde{b} \in \mathbb{G}\big)$ and $q \in \mathsf{poly}(\lambda)$ tuples $(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7$ such that*

$$e(a,\tilde{a}) = e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j) \quad \text{and} \quad e(b,\tilde{b}) = e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \tag{1}$$

to find a new tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ satisfying relation (1) and such that $z^\star \neq 1_{\mathbb{G}}$ and $z^\star \neq z_j$ for $j \in \{1, \ldots, q\}$.

2.2 Groth-Sahai Proof Systems

In the following notations, for equal-dimension vectors or matrices A and B containing group elements, $A \odot B$ stands for their entry-wise product.

In their instantiations based on the DLIN assumption, the Groth-Sahai (GS) techniques [40] make use of prime order groups and a common reference string comprising vectors $\vec{f}_1, \vec{f}_2, \vec{f}_3 \in \mathbb{G}^3$, where $\vec{f}_1 = (f_1, 1, g)$, $\vec{f}_2 = (1, f_2, g)$ for some $f_1, f_2 \in \mathbb{G}$. To commit to an element $X \in \mathbb{G}$, one sets $\vec{C} = (1, 1, X) \odot \vec{f}_1^{\,r} \odot \vec{f}_2^{\,s} \odot \vec{f}_3^{\,t}$ with $r, s, t \xleftarrow{R} \mathbb{Z}_p^*$. When the CRS is configured to give perfectly sound proofs, we have $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$ where $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. Commitments $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ are then Boneh-Boyen-Shacham (BBS) ciphertexts [16] that can be decrypted using $\beta_1 = \log_g(f_1)$, $\beta_2 = \log_g(f_2)$. In the witness indistinguishability (WI) setting, vectors $\vec{f}_1, \vec{f}_2, \vec{f}_3$ are linearly independent and $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of CRS are computationally indistinguishable.

To commit to a scalar $x \in \mathbb{Z}_p$, one computes $\vec{C} = \vec{\varphi}^x \odot \vec{f}_1^{\,r} \odot \vec{f}_2^{\,s}$, where $r, s \xleftarrow{R} \mathbb{Z}_p^*$, using a CRS comprising vectors $\vec{\varphi}, \vec{f}_1, \vec{f}_2$. In the soundness setting, $\vec{\varphi}, \vec{f}_1, \vec{f}_2$ are linearly independent (typically $\vec{\varphi} = \vec{f}_3 \odot (1, 1, g)$ where $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$) whereas, in the WI setting, choosing $\vec{\varphi} = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$ gives a perfectly hiding commitment since $\vec{C}$ is always a BBS encryption of $1_{\mathbb{G}}$, no matter which exponent x is committed to.

To prove that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element (made of a constant number of group elements) per relation.

Such proofs are available for pairing-product equations, which are relations of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{2}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$. Efficient proofs also exist for multi-exponentiation equations

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T, \tag{3}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{G}$, for $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

In pairing-product equations, proofs for quadratic equations require 9 group elements whereas linear equations (*i.e.*, where $a_{ij} = 0$ for all i, j in equation (2)) only take 3 group elements each. Linear multi-exponentiation equations of the type $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$ demand 2 group elements.

Multi-exponentiation equations admit zero-knowledge (NIZK) proofs at no additional cost. On a simulated CRS (prepared for the WI setting), a trapdoor makes it is possible to simulate proofs without knowing witnesses and simulated proofs have the same distribution as real proofs. In contrast, pairing-product equations do not always have NIZK proofs. Fortunately, NIWI proofs will be sufficient here.

2.3 Structure-Preserving Signatures

Several applications (see [2, 3, 36, 31, 4] for examples) require to sign groups elements while preserving the feasibility of efficiently proving that a committed signature is valid for a committed group element.

In [2, 3], Abe, Haralambiev and Ohkubo showed how to conveniently sign n group elements at once using signatures consisting of $O(1)$ group elements. Their scheme (which is referred to as the AHO signature in the paper) makes use of bilinear groups of prime order. In the context of symmetric pairings, the description below assumes public parameters $\mathsf{pp} = \big((\mathbb{G}, \mathbb{G}_T),\ g\big)$ consisting of groups $(\mathbb{G}, \mathbb{G}_T)$ of order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ is a security parameter, with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and a generator $g \in \mathbb{G}$.

Keygen(pp, n)**:** given an upper bound $n \in \mathbb{N}$ on the number of group elements that can be signed altogether, choose generators $G_r, H_r \xleftarrow{R} \mathbb{G}$. Pick $\gamma_z, \delta_z \xleftarrow{R} \mathbb{Z}_p$ and $\gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$, for $i = 1$ to n. Then, compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$ and $G_i = G_r^{\gamma_i}$, $H_i = H_r^{\delta_i}$ for each $i \in \{1, \ldots, n\}$. Finally, choose exponents $\alpha_a, \alpha_b \xleftarrow{R} \mathbb{Z}_p$ and define $A = e(G_r, g^{\alpha_a})$ and $B = e(H_r, g^{\alpha_b})$. Set the public key as

$$pk = \big(G_r,\ H_r,\ G_z,\ H_z,\ \{G_i, H_i\}_{i=1}^n,\ A,\ B\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$$

while the private key consists of $sk = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign$(sk, (M_1, \ldots, M_n))$**:** to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $sk = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$, choose $\zeta, \rho, \tau, \nu, \omega \xleftarrow{R} \mathbb{Z}_p$ and compute $\theta_1 = g^\zeta$ as well as

$$\theta_2 = g^{\rho - \gamma_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}, \qquad \theta_3 = G_r^\tau, \qquad \theta_4 = g^{(\alpha_a - \rho)/\tau},$$

$$\theta_5 = g^{\nu - \delta_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}, \qquad \theta_6 = H_r^\omega, \qquad \theta_7 = g^{(\alpha_b - \nu)/\omega},$$

The signature consists of $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$.

Verify$(pk, \sigma, (M_1, \ldots, M_n))$**:** parse σ as $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \in \mathbb{G}^7$ and return 1 iff these equalities hold:

$$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^{n} e(G_i, M_i), \tag{4}$$

$$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^{n} e(H_i, M_i). \tag{5}$$

In [2, 3], the scheme was proved to be existentially unforgeable under chosen-message attacks under the q-SFP assumption, where q is the maximal number of signing queries.

Abe *et al.* [2, 3] also showed that signatures can be publicly randomized to obtain a different signature $\{\theta'_i\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk, \sigma)$ on $(M_1, \ldots, M_n)$. After randomization, we have $\theta'_1 = \theta_1$ while $\{\theta'_i\}_{i=2}^7$ are uniformly distributed among the values satisfying the equalities $e(G_r, \theta'_2) \cdot e(\theta'_3, \theta'_4) = e(G_r, \theta_2) \cdot e(\theta_3, \theta_4)$ and $e(H_r, \theta'_5) \cdot e(\theta'_6, \theta'_7) = e(H_r, \theta_5) \cdot e(\theta_6, \theta_7)$. Moreover, $\{\theta'_i\}_{i\in\{3,4,6,7\}}$ are statistically independent of $(M_1, \ldots, M_n)$ and the rest of the signature. This implies that, in anonymity-related protocols, re-randomized $\{\theta'_i\}_{i\in\{3,4,6,7\}}$ can be safely revealed as long as $(M_1, \ldots, M_n)$ and $\{\theta'_i\}_{i\in\{1,2,5\}}$ are given in committed form.

In [4], Abe, Groth, Haralambiev and Ohkubo described a more efficient structure-preserving signature based on interactive assumptions. Here, we use the scheme of [2, 3] so as to rely on falsifiable assumptions.

2.4 The NNL Framework for Broadcast Encryption

The Subset Cover framework [51] considers secret-key broadcast encryption schemes with $N = 2^\ell$ registered receivers. Each one of them is associated with a leaf of a complete binary tree T of height ℓ and each tree node is assigned a secret key. If $\mathcal{N}$ denotes the universe of users and $\mathcal{R} \subset \mathcal{N}$ is the set of revoked receivers, the idea of the framework is to partition the set of non-revoked users into m disjoint subsets $S_1, \ldots, S_m$ such that $\mathcal{N}\backslash\mathcal{R} = S_1 \cup \ldots \cup S_m$. Depending on the way to partition $\mathcal{N}\backslash\mathcal{R}$ and the distribution of keys to users, different instantiations and tradeoffs are possible.

The Complete Subtree Method. In this technique, each subset S_i consists of the leaves of a complete subtree rooted at some node x_i of T. Upon registration, each user obtains secret keys for all nodes on the path connecting his leaf to the root of T (and thus $O(\ell)$ keys overall). By doing so, users in $\mathcal{N}\backslash\mathcal{R}$ can decrypt the content if the latter is enciphered using symmetric keys $K_1, \ldots, K_m$ corresponding to the roots of subtrees $S_1, \ldots, S_m$. As showed in [51], the CS partitioning method entails at most $m \leq r \cdot \log(N/r)$ subsets, where $r = |\mathcal{R}|$. Each transmission requires to send $O(r \cdot \log N)$ symmetric encryptions while, at each user, the storage complexity is $O(\log N)$.

As noted in [51, 34], a single-level identity-based encryption scheme allows implementing a public-key variant of the CS method. The master public key of the IBE scheme forms the public key of the broadcast encryption system, which allows for public keys of size $O(1)$ (instead of $O(N)$ in instantiations using ordinary public-key encryption). When users join the system, they obtain $O(\ell)$ IBE private keys (in place of symmetric keys) associated with the "identities" of nodes on the path between their leaf and the root.

The Subset Difference Method. The SD method reduces the transmission cost to $O(r)$ at the expense of increased storage requirements. For each node $x_j \in \mathsf{T}$, we call T_{x_j} the subtree rooted at x_j. The set $\mathcal{N}\backslash\mathcal{R}$ is now divided into disjoint subsets $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$. For each $i \in \{1, \ldots, m\}$, the subset S_{k_i,u_i} is determined by a node x_{k_i} and one of its descendants x_{u_i} – which are called *primary* and *secondary* roots of S_{k_i,u_i}, respectively – and it consists of the leaves of $\mathsf{T}_{x_{k_i}}$ that are not in $\mathsf{T}_{x_{u_i}}$. Each user thus belongs to much more generic subsets than in the CS method and this allows reducing the maximal number of subsets to $m = 2r - 1$ (see [51] for a proof of this bound).

A more complex key distribution is necessary to avoid a prohibitive storage overhead. Each subset S_{k_i,u_i} is assigned a "proto-key" $P_{x_{k_i},x_{u_i}}$ that allows deriving the actual symmetric encryption key K_{k_i,u_i} for S_{k_i,u_i} and as well as proto-keys $P_{x_{k_i},x_{u_l}}$ for any descendant x_{u_l} of x_{u_i}. At the same time, $P_{x_{k_i},x_{u_l}}$ should be hard to compute without a proto-key $P_{x_{k_i},x_{u_i}}$ for an ancestor x_{u_i} of x_{u_l}. The key distribution phase then proceeds as follows. Let user i be assigned a leaf v_i and let $\epsilon = x_0, x_1, \ldots, x_\ell = v_i$ denote the path from the root ϵ to v_i. For each subtree T_{x_j} (with $j \in \{1, \ldots, \ell\}$), if copath_{x_j} denotes the set of all siblings of nodes on the path from x_j to v_i, user i must obtain proto-keys $P_{x_j,w}$ for each node $w \in \mathsf{copath}_{x_j}$ because he belongs to the generic subset whose primary root is x_j and whose secondary root is w. By storing $O(\ell^2)$ proto-keys (*i.e.*, $O(\ell)$ for each subtree T_{x_j}), users will be able to derive keys for all generic subsets they belong to.

In [34], Dodis and Fazio extended the SD method to the public-key setting using hierarchical identity-based encryption. In the tree, each node w at depth $\leq \ell$ has a label $\langle w \rangle$ which is defined by assigning the label ε to the root (at depth 0). The left and right children of w are then labeled with $\langle w \rangle || 0$ and $\langle w \rangle || 1$, respectively. For each subset S_{k_i,u_i} of $\mathcal{N} \backslash \mathcal{R}$, the sender considers the primary and secondary roots x_{k_i}, x_{u_i} and parses the label $\langle x_{u_i} \rangle$ as $\langle x_{k_i} \rangle || u_{i,\ell_{i,1}} \ldots u_{i,\ell_{i,2}}$, with $u_{i,j} \in \{0,1\}$ for each $j \in \{\ell_{i,1}, \ldots, \ell_{i,2}\}$. Then, he computes a HIBE ciphertext for the hierarchical identity $(\langle x_{k_i} \rangle, u_{i,\ell_{i,1}}, \ldots, u_{i,\ell_{i,2}})$ at level $\ell_{i,2} - \ell_{i,1} + 2$. Upon registration, if $\epsilon = x_0, \ldots, x_\ell = v_i$ denotes the path from the root to his leaf v_i, for each subtree T_{x_j}, user i receives exactly one HIBE private key for each $w \in \mathsf{copath}_{x_j}$: namely, for each $w \in \mathsf{copath}_{x_j}$, there exist $\ell_1, \ell_2 \in \{1, \ldots, \ell\}$ such that $\langle w \rangle = \langle x_j \rangle || w_{\ell_1} \ldots w_{\ell_2}$ with $w_j \in \{0,1\}$ for all $j \in \{\ell_1, \ldots, \ell_2\}$ and user i obtains a HIBE private key for the hierarchical identity $(\langle x_j \rangle, w_{\ell_1}, \ldots, w_{\ell_2})$. By construction, this key will allow user i to decrypt any HIBE ciphertext encrypted for a subset whose primary root is x_j and whose secondary root is a descendant of w. Overall, each user thus has to store $O(\log^2 N)$ HIBE private keys.

2.5 Revocable Group Signatures

We consider group signature schemes that have their lifetime divided into revocation epochs at the beginning of which group managers update their revocation lists.

The syntax and the security model are similar to [49] but they build on those defined by Kiayias and Yung [45]. Like the Bellare-Shi-Zhang model [11], the latter assumes an interactive join protocol between the group manager and the prospective user. This protocol provides the user with a membership certificate and a membership secret. Such protocols may consist of several rounds of interaction.

Syntax. We denote by $N \in \mathsf{poly}(\lambda)$ the maximal number of group members. At the beginning of each revocation epoch t, the group manager publicizes an up-to-date revocation list RL_t and we denote by $\mathcal{R}_t \subset \{1, \ldots, N\}$ the corresponding set of revoked users (we assume that $\mathcal{R}_t$ is part of RL_t). A revocable group signature (R-GS) scheme consists of the following algorithms or protocols.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and a maximal number of group members $N \in \mathbb{N}$, this algorithm (which is run by a trusted party) generates a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Keys $\mathcal{S}_{\mathsf{GM}}$ and $\mathcal{S}_{\mathsf{OA}}$ are given to the appropriate authority while $\mathcal{Y}$ is publicized. The algorithm also initializes a public state St comprising a set data structure $St_{users} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$.

Join: is an interactive protocol between the group manager GM and a user $\mathcal{U}_i$ where the latter becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take as input $\mathcal{Y}$. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$, terminates with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol successfully terminates, the group manager updates the public state St by setting $St_{users} := St_{users} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.

Revoke: is a (possibly randomized) algorithm allowing the GM to generate an updated revocation list RL_t for the new revocation epoch t. It takes as input a public key $\mathcal{Y}$ and a set $\mathcal{R}_t \subset St_{users}$ that identifies the users to be revoked. It outputs an updated revocation list RL_t for epoch t.

Sign: given a revocation epoch t with its revocation list RL_t, a membership certificate cert_i, a membership secret sec_i and a message M, this algorithm outputs $\perp$ if $i \in \mathcal{R}_t$ and a signature σ otherwise.

Verify: given a signature σ, a revocation epoch t, the corresponding revocation list RL_t, a message M and a group public key $\mathcal{Y}$, this deterministic algorithm returns either 0 or 1.

Open: takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$ for the indicated revocation epoch t, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{users} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

Each membership certificate contains a unique tag that identifies the user.

A R-GS scheme must satisfy three security notions defined in appendix A. The first one is called *security against misidentification attacks*. It requires that, even if the adversary can introduce and revoke users at will, it cannot produce a signature that traces outside the set of unrevoked adversarially-controlled users.

As in ordinary (*i.e.*, non-revocable) group signatures, the notion of *security against framing attacks* mandates that, even if the whole system colludes against a user, that user will not bear responsibility for messages that he did not sign. Finally, the notion of *anonymity* is also defined (in the presence of a signature opening oracle) as in the models of [11, 45].

3 A Revocable Group Signature Based on the Subset Difference Method

As already mentioned, the idea is to turn the NNL global ciphertext into a revocation list in the group signature. Each group member is associated with a leaf of a binary tree of height ℓ and the outcome of the join protocol is the user obtaining a membership certificate that contains the same key material as in the public-key variant of the SD method (*i.e.*, $O(\ell^2)$ HIBE private keys). To ensure traceability and non-frameability, these NNL private keys are linked to a group element X, that only the user knows the discrete logarithm of, by means of structure-preserving signatures.

At each revocation epoch t, the group manager generates an up-to-date revocation list RL_t consisting of $O(r)$ HIBE ciphertexts, each of which is signed using a structure-preserving signature. When it comes to sign a message, the user $\mathcal{U}_i$ proves that he is not revoked by providing evidence that he is capable of decrypting one of the HIBE ciphertexts in RL_t. To this end, $\mathcal{U}_i$ commits to that HIBE ciphertext C_l and proves that he holds a key that decrypts C_l. To convince the verifier that C_l belongs to RL_t, he proves knowledge of a signature on the committed HIBE ciphertext C_l (this technique is borrowed from the set membership and range proofs of [57, 25]). Of course, to preserve the anonymity of signers, we need a HIBE scheme with constant-size ciphertexts (otherwise, the length of the committed ciphertext could betray the signer's location in the tree), which is why the Boneh-Boyen-Goh construction [15] is the ideal candidate.

The scheme is made anonymous and non-frameable using the same techniques as Groth [39] in steps 4-6 of the signing algorithm. As for the security against misidentification attacks, we cannot prove it by relying on the standard collusion-resistance (captured by Definition 7 in appendix B.1) of the HIBE scheme. In the proof of Theorem 1, the problem appears in the treatment of forgeries that open to a revoked user: while this user cannot have obtained a private key that decrypts the committed HIBE ciphertext of the forgery (because he is revoked), unrevoked adversarially-controlled users can. To solve this problem, we need to rest on a non-standard security property (formalized by Definition 8 in appendix B.1) called "key-robustness". This notion asks that, given a private key generated for some hierarchical identity using specific random coins, it be infeasible to compute the private key of a different identity for the *same random coins* and even *knowing* the *master secret key* of the HIBE scheme. While unusual, this property can be proved (by Lemma 1 in appendix B.2) under the standard Diffie-Hellman assumption for the BBG construction.

Perhaps surprisingly, even though we rely on the BBG HIBE, we do not need its underlying q-type assumption [15]. The reason is that the master secret key of the scheme is unnecessary here as its role is taken over by the private key of a structure-preserving signature. In the ordinary BBG system (recalled in appendix B.2), private keys contain components of the form $(g_2^{\alpha} \cdot F(\mathsf{ID})^r, g^r)$, for some $r \in \mathbb{Z}_p$, where g_2^{α} is the master secret key and $F(\mathsf{ID})$ is a function of the hierarchical identity. In the join protocol, the master key g_2^{α} disappears: the user obtains a private key of the form $(F(\mathsf{ID})^r, g^r)$ and an AHO signature is used to bind the user's membership public key X to g^r. The latter can be thought of as a public key for a one-time variant (the one-time nature is what allows for a proof of selective-message security in the standard model) of the Boneh-Lynn-Shacham signature [17]. The underlying one-time private key $r \in \mathbb{Z}_p$

is used to compute $F(\mathsf{ID})^r$ as well as a number of delegation components allowing to derive signatures for messages that ID is a prefix of (somewhat in the fashion of wildcard signatures [1][Section 6]).

3.1 Construction

As in Section 2.4, $\langle x \rangle$ denotes the label of node $x \in \mathsf{T}$ and, for any sub-tree T_{x_j} rooted at x_j and any leaf v_i of T_{x_j}, copath_{x_j} denotes the set of all siblings of nodes on the path from x_j to v_i, not counting x_j itself.

As is standard in group signatures, the description below assumes that, before joining the group, user $\mathcal{U}_i$ chooses a long term key pair $(\mathsf{usk}[i], \mathsf{upk}[i])$ and registers it in some PKI.

$\mathsf{Setup}(\lambda, N)$: given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N = 2^\ell$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \xleftarrow{R} \mathbb{G}$.
2. Generate two key pairs $(sk^{(0)}_{\mathsf{AHO}}, pk^{(0)}_{\mathsf{AHO}})$ and $(sk^{(1)}_{\mathsf{AHO}}, pk^{(1)}_{\mathsf{AHO}})$ for the AHO signature in order to sign messages of two group elements. These key pairs consist of

$$pk^{(d)}_{\mathsf{AHO}} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^2,\ A^{(d)},\ B^{(d)}\Big)$$

   and $sk^{(d)}_{\mathsf{AHO}} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^2\big)$, where $d \in \{0,1\}$.
3. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ s.t. $\vec{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f}_2 = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \cdot \vec{f}_2^{\,\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \xleftarrow{R} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$.
4. Choose $(U, V) \xleftarrow{R} \mathbb{G}^2$ that, together with f_1, f_2, g, will form a public encryption key.
5. Generate a master public key mpk_{BBG} for the Boneh-Boyen-Goh HIBE. Such a public key consists[3] of $mpk_{\mathsf{BBG}} = \big(\ \{h_i\}_{i=0}^{\ell}\big)$, where $\ell = \log_2(N)$, and no master secret key is needed.
6. Select an injective encoding[4] function $\mathcal{H} : \{0,1\}^{\leq \ell} \to \mathbb{Z}_p^*$ and a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and the group public key is

$$\mathcal{Y} := \Big(g,\ pk^{(0)}_{\mathsf{AHO}},\ pk^{(1)}_{\mathsf{AHO}},\ mpk_{\mathsf{BBG}},\ \mathbf{f},\ (U, V),\ \mathcal{H},\ \Sigma\Big).$$

$\mathsf{Join}^{(\mathrm{GM}, \mathcal{U}_i)}$: the GM and the prospective user $\mathcal{U}_i$ run the following protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ picks $x \xleftarrow{R} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})$. If the value X already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} aborts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v_i of label $\langle v_i \rangle = v_{i,1} \ldots v_{i,\ell} \in \{0,1\}^\ell$ in the tree T. Let $x_0 = \epsilon$, $x_1, \ldots, x_{\ell-1}, x_\ell = v_i$ be the path from v_i to the root ϵ of T. For $j = 0$ to ℓ, J_{GM} does the following.
   a. Consider the sub-tree T_{x_j} rooted at node x_j. Let copath_{x_j} be the co-path from x_j to v_i.
   b. For each node $w \in \mathsf{copath}_{x_j}$, since x_j is an ancestor of w, $\langle x_j \rangle$ is a prefix of $\langle w \rangle$ and we denote by $w_{\ell_1} \ldots w_{\ell_2} \in \{0,1\}^{\ell_2 - \ell_1 + 1}$, for some $\ell_1 \leq \ell_2 \leq \ell$, the suffix of $\langle w \rangle$ coming right after $\langle x_j \rangle$.

[3] As mentioned earlier, in comparison with the original HIBE scheme (recalled in appendix B.2) where mpk_{BBG} includes $(g_1 = g^\alpha, g_2)$ and $msk_{\mathsf{BBG}} = g_2^\alpha$, the public elements g_1 and g_2 have disappeared.

[4] This encoding allows making sure that "identities" will be non-zero at each level. Since the set $\{0,1\}^{\leq \ell}$ is of cardinality $\sum_{i=0}^{\ell} 2^i = 2^{\ell+1} - 1 < p - 1$, such a function can be efficiently constructed without any intractability assumption.

b.1 Choose a random $r \xleftarrow{R} \mathbb{Z}_p$ and compute a HIBE private key

$$\begin{aligned} d_w &= (D_{w,1}, D_{w,2}, K_{w,\ell_2-\ell_1+3}, \ldots, K_{w,\ell}) \\ &= \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_j \rangle)} \cdot h_2^{\mathcal{H}(w_{\ell_1})} \cdots h_{\ell_2-\ell_1+2}^{\mathcal{H}(w_{\ell_2})}\big)^r,\ g^r,\ h_{\ell_2-\ell_1+3}^r, \ldots,\ h_\ell^r \Big) \end{aligned}$$

for the identity $(\mathcal{H}(\langle x_j \rangle), \mathcal{H}(w_{\ell_1}), \ldots, \mathcal{H}(w_{\ell_2})) \in (\mathbb{Z}_p^*)^{\ell_2-\ell_1+2}$.

b.2 Using $sk_{\mathsf{AHO}}^{(0)}$, generate an AHO signature $\sigma_w = (\theta_{w,1}, \ldots, \theta_{w,7})$ on $(X, D_{w,2}) \in \mathbb{G}^2$ so as to bind the HIBE private key d_w to the value X that identifies $\mathcal{U}_i$.

3. J_{GM} sends $\langle v_i \rangle \in \{0,1\}^\ell$, and the HIBE private keys $\{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ that verifies their validity. If these keys are all well-formed, $\mathsf{J}_{\mathsf{user}}$ acknowledges them by generating a digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X || \{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}\big)$ and sends it back to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big(X || \{\{d_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} returns the AHO signatures $\{\{\sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ and stores the conversation transcript $\mathsf{transcript}_i = (X, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i)$ in the database St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate cert_i as $\mathsf{cert}_i = \big(\langle v_i \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X\big)$, where X will serve as the tag that identifies $\mathcal{U}_i$. The membership secret sec_i is defined to be $\mathsf{sec}_i = x$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:**

1. Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} := \big(sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)}\big)$.
2. Using the SD covering algorithm, find a cover of the unrevoked user set $\{1, \ldots, N\} \backslash \mathcal{R}_t$ as the union of disjoint subsets of the form $S_{k_1,u_1}, \ldots, S_{k_m,u_m}$, with $m \leq 2 \cdot |\mathcal{R}_t| - 1$.
3. For $i = 1$ to m, do the following.

   a. Consider S_{k_i,u_i} as the difference between sub-trees rooted at an internal node x_{k_i} and one of its descendants x_{u_i}. The label of x_{u_i} can be written $\langle x_{u_i} \rangle = \langle x_{k_i} \rangle || u_{i,\ell_{i,1}} \ldots u_{i,\ell_{i,2}}$ for some $\ell_{i,1} < \ell_{i,2} \leq \ell$ and where $u_{i,\kappa} \in \{0,1\}$ for each $\kappa \in \{\ell_{i,1}, \ldots, \ell_{i,2}\}$. Then, compute an encoding of S_{k_i,u_i} as a group element

   $$C_i = h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_i} \rangle)} \cdot h_2^{\mathcal{H}(u_{i,\ell_{i,1}})} \cdots h_{\ell_{i,2}-\ell_{i,1}+2}^{\mathcal{H}(u_{i,\ell_{i,2}})}.$$

   Note that C_i can be thought of as a de-randomized HIBE ciphertext for the hierarchical identity $\big(\mathcal{H}(\langle x_{k_i} \rangle), \mathcal{H}(u_{i,\ell_{i,1}}), \ldots, \mathcal{H}(u_{i,\ell_{i,2}})\big) \in (\mathbb{Z}_p^*)^{\ell_{i,2}-\ell_{i,1}+2}$.

   b. To authenticate the HIBE ciphertext C_i and bind it to the revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to generate an AHO signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on the pair $(C_i, g^t) \in \mathbb{G}^2$, where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return the revocation data RL_t which is defined to be

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\langle x_{k_i} \rangle,\ \langle x_{u_i} \rangle,\ \big(C_i, \Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7})\big)\}_{i=1}^{m}\Big) \tag{6}$$

Sign$(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generate a one-time signature key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i as $\big(\langle v_i \rangle, \{\{(d_w, \sigma_w)\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X\big)$ and sec_i as $x \in \mathbb{Z}_p$. Then, $\mathcal{U}_i$ conducts the following steps.

1. Using RL_t, determine the set S_{k_l,u_l}, with $l \in \{1, \ldots, m\}$, that contains the leaf v_i (this subset must exist since $i \notin \mathcal{R}_t$) and let x_{k_l} and x_{u_l} denote the primary and secondary roots of S_{k_l,u_l}. Since

x_{k_l} is an ancestor of x_{u_l}, we can write $\langle x_{u_l}\rangle = \langle x_{k_l}\rangle || u_{l,\ell_1} \ldots u_{l,\ell_2}$, for some $\ell_1 < \ell_2 \leq \ell$ and with $u_{l,\kappa} \in \{0,1\}$ for each $\kappa \in \{\ell_1, \ldots, \ell_2\}$. The signer $\mathcal{U}_i$ computes a HIBE decryption key of the form

$$(D_{l,1}, D_{l,2}) = \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_l}\rangle)} \cdot h_2^{\mathcal{H}(u_{l,\ell_1})} \cdots h_{\ell_2-\ell_1+2}^{\mathcal{H}(u_{l,\ell_2})}\big)^r,\ g^r \Big). \tag{7}$$

This is possible since, if we denote by $\langle x_{k,l}\rangle || u_{l,\ell_1} \ldots u_{l,\ell_1'}$ the shortest prefix of $\langle x_{u_l}\rangle$ that is not a prefix of $\langle v_i\rangle$, the key material $\{d_w\}_{w \in \mathsf{copath}_{x_{k_l}}}$ corresponding to the sub-tree rooted at x_{k_l} contains a HIBE private key $d_w = (D_{w,1}, D_{w,2}, K_{w,\ell_1'-\ell_1+3}, \ldots, K_{w,\ell})$ such that

$$d_w = \Big(\big(h_0 \cdot h_1^{\mathcal{H}(\langle x_{k_l}\rangle)} \cdot h_2^{\mathcal{H}(u_{l,\ell_1})} \cdots h_{\ell_1'-\ell_1+2}^{\mathcal{H}(u_{l,\ell_1'})}\big)^r,\ g^r,\ h_{\ell_1'-\ell_1+3}^r, \ldots, h_\ell^r \Big),$$

which allows deriving a key of the form (7) for the same $r \in \mathbb{Z}_p$ (*i.e.*, $D_{l,2} = D_{w,2}$).

2. To prove his ability to "decrypt" C_l, $\mathcal{U}_i$ first re-randomizes Θ_l as $\{\Theta'_{l,i}\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(1)}, \Theta_l)$. Then, he computes a Groth-Sahai commitment com_{C_l} to C_l as well as commitments $\{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}$ to $\{\Theta'_{l,i}\}_{i\in\{1,2,5\}}$. He generates a proof π_{C_l} that C_l is a certified HIBE ciphertext for epoch t: *i.e.*, π_{C_l} provides evidence that

$$\begin{aligned} A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_2^{(1)}, g^t)^{-1} &= e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot e(G_1^{(1)}, C_l), \\ B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_2^{(1)}, g^t)^{-1} &= e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot e(H_1^{(1)}, C_l), \end{aligned} \tag{8}$$

Then, $\mathcal{U}_i$ generates commitments $\{com_{D_{l,i}}\}_{i=1}^2$ to the HIBE key components $\{D_{l,i}\}_{i=1}^2$ derived at step 1 and computes a proof π_{D_l} that $e(D_{l,1}, g) = e(C_l, D_{l,2})$. The latter is quadratic and requires 9 group elements. Since $\{\Theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ are constants, equations (8) are linear and require 3 elements each. Hence, π_{C_l} and π_{D_l} take 15 elements altogether.

3. Let $\sigma_l = (\theta_{l,1}, \ldots, \theta_{l,7})$ be the AHO signature on $(X, D_{l,2})$. Compute $\{\theta'_{l,i}\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}^{(0)}, \sigma_l)$ and generate commitments $\{com_{\theta'_{l,i}}\}_{i\in\{1,2,5\}}$ to $\{\theta'_{l,i}\}_{i\in\{1,2,5\}}$ as well as a commitment com_X to X. Then, generate a proof π_{σ_l} that committed variables satisfy the verification equations

$$\begin{aligned} A^{(0)} \cdot e(\theta'_{l,3}, \theta'_{l,4})^{-1} &= e(G_z^{(0)}, \theta'_{l,1}) \cdot e(G_r^{(0)}, \theta'_{l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, D_{l,2}), \\ B^{(0)} \cdot e(\theta'_{l,6}, \theta'_{l,7})^{-1} &= e(H_z^{(0)}, \theta_{l,1}) \cdot e(H_r^{(0)}, \theta'_{l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, D_{l,2}) \end{aligned}$$

Since these equations are linear, π_{σ_l} requires 6 group elements.

4. Using VK as a tag (we assume that it is first hashed onto $\mathbb{Z}_p$ in such a way that it can be interpreted as a $\mathbb{Z}_p$ element), compute a tag-based encryption [47] of X by drawing $z_1, z_2 \xleftarrow{R} \mathbb{Z}_p$ and setting

$$(\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$

5. Generate a NIZK proof that $com_X = (1,1,X) \cdot \vec{f}_1^{\,\phi_{X,1}} \cdot \vec{f}_2^{\,\phi_{X,2}} \cdot \vec{f}_3^{\,\phi_{X,3}}$ and (Ψ_1, Ψ_2, Ψ_3) are BBS encryptions of the same value X. If we write $\vec{f}_3 = (f_{3,1}, f_{3,2}, f_{3,3})$, the Groth-Sahai commitment com_X can be written as $(f_1^{\phi_{X,1}} \cdot f_{3,1}^{\phi_{X,3}}, f_2^{\phi_{X,2}} \cdot f_{3,2}^{\phi_{X,3}}, X \cdot g^{\phi_{X,1}+\phi_{X,2}} \cdot f_{3,3}^{\phi_{X,3}})$, so that we have

$$com_X \odot (\Psi_1, \Psi_2, \Psi_3)^{-1} = \big(f_1^{\tau_1} \cdot f_{3,1}^{\tau_3},\ f_2^{\tau_2} \cdot f_{3,2}^{\tau_3},\ g^{\tau_1+\tau_2} \cdot f_{3,3}^{\tau_3}\big) \tag{9}$$

with $\tau_1 = \phi_{X,1} - z_1$, $\tau_2 = \phi_{X,2} - z_2$, $\tau_3 = \phi_{X,3}$. The signer $\mathcal{U}_i$ commits to $\tau_1, \tau_2, \tau_3 \in \mathbb{Z}_p$ (by computing $com_{\tau_j} = \vec{\varphi}^{\tau_j} \cdot \vec{f}_1^{\,\phi_{\tau_j,1}} \cdot \vec{f}_2^{\,\phi_{\tau_j,2}}$, for $j \in \{1,2,3\}$, using the vector $\vec{\varphi} = \vec{f}_3 \cdot (1,1,g)$ and random $\{\phi_{\tau_j,1}, \phi_{\tau_j,2}\}_{j=1}^3$), and generates proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that τ_1, τ_2, τ_3 satisfy the three relations (9). Since these are linear equations, proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ cost 2 elements each.

6. Compute $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ and generate a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof that committed variables σ_{VK} and X satisfy $e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g,g)$. This relation is quadratic and requires a proof consisting of 9 group elements. We denote this proof by $\pi_{\sigma_{\mathsf{VK}}} = (\vec{\pi}_{\sigma_{\mathsf{VK}},1}, \vec{\pi}_{\sigma_{\mathsf{VK}},2}, \vec{\pi}_{\sigma_{\mathsf{VK}},3})$.
7. Compute $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ and

$$\mathbf{com} = \big(com_{C_l}, \{com_{D_{l,i}}\}_{i=1}^{2}, com_X, \{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\tau_i}\}_{i=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big)$$
$$\mathbf{\Pi} = (\pi_{C_l}, \pi_{D_l}, \pi_{\sigma_l}, \pi_{eq\text{-}com,1}, \pi_{eq\text{-}com,2}, \pi_{eq\text{-}com,3}, \pi_{\sigma_{\mathsf{VK}}})$$

Return the signature $\sigma = \big(\mathsf{VK}, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big)$.

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above and do the following.

1. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0.
2. Return 0 if $e(\Psi_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Psi_4)$ or $e(\Psi_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Psi_5)$.
3. Return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, parse the signature σ as above and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, compute $\tilde{X} = \Psi_3 \cdot \Psi_1^{-1/\beta_1} \cdot \Psi_2^{-1/\beta_2}$. In the database St_{trans}, find a record $\langle i, \mathsf{transcript}_i = (X, \{\{d_w, \sigma_w\}_{w\in\mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, sig_i)\rangle$ such that $X = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

From an efficiency point of view, for each $i \in \{1\ldots, m\}$, RL_t comprises 8 group elements plus the labels of nodes that identify S_{k_i,u_i}. If $\lambda_{\mathbb{G}}$ denotes the bitlength of a group element, the number of bits of RL_t is thus bounded by $2 \cdot |\mathcal{R}_t| \cdot (8 \cdot \lambda_{\mathbb{G}} + 2\log N) < 2 \cdot |\mathcal{R}_t| \cdot (9\lambda_{\mathbb{G}})$ bits (as $\log N < \lambda_{\mathbb{G}}/2$ since $\lambda \leq \lambda_{\mathbb{G}}$ and N is polynomial). The size of revocation lists thus amounts to that of at most $18 \cdot |\mathcal{R}_t|$ group elements.

Group members need $O(\log^3 N)$ group elements to store their membership certificate. As far as the size of signatures goes, **com** and $\mathbf{\Pi}$ require 42 and 36 group elements, respectively. If the one-time signature of [38] is used, σ consists of 96 group elements, which is less than twice the size of Groth's signatures [39]. At the 128-bit security level, if each element has a representation of 512 bits, a signature takes 6 kB.

Verifying signatures takes constant time. The cost of each signature generation is dominated by at most $\ell = \log N$ exponentiations to derive a HIBE private key at step 1. However, this step only has to be executed once per revocation epoch, at the first signature of that epoch.

3.2 Security

Theorem 1 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP problem is hard for $q = \max(\ell^2 \cdot q_a, q_r^2)$, where q_a and q_r denote the maximal numbers of $Q_{\mathsf{a\text{-}join}}$ queries and Q_{revoke} queries, respectively, and $\ell = \log N$.*

Proof. To mount a successful misidentification attack, the adversary $\mathcal{A}$ must output a non-trivial signature for which the opening algorithm fails to point to an unrevoked adversarially-controlled group member.

Let $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ denote $\mathcal{A}$'s forgery and parse $\mathbf{com}^\star$ as

$$\mathbf{com}^\star = \Big(com_C^\star, \{com_{D_i}^\star\}_{i=1}^{2}, com_X^\star, \{com_{\Theta'_i}^\star\}_{i\in\{1,2,5\}}, \{com_{\theta'_i}^\star\}_{i\in\{1,2,5\}}, \{com_{\tau_i}^\star\}_{i=1}^{3}, com_{\sigma_{\mathsf{VK}^\star}}^\star\Big).$$

By hypothesis, it must be the case that $\mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St) \notin U^a \backslash \mathcal{R}_{t^\star}$, where U^a denotes the set of adversarially-controlled users. Depending on extractable commitments $com_X^\star$, $com_C^\star$, $\{com_{D_i}^\star\}_{i=1}^{2}$, $\{com_{\Theta'_i}^\star\}_{i\in\{1,2,5\}}$, $\{com_{\theta'_i}^\star\}_{i\in\{1,2,5\}}$ and their contents, we distinguish the following cases:

- **Type I forgeries** are those for which $com^{\star}_{C_l}$ contains a group element $C^{\star}$ such that $(C^{\star}, g^{t^{\star}})$ was never signed when the latest revocation list $RL_{t^{\star}}$ was generated.
- **Type II forgeries** are such that $com^{\star}_{C}$ contains a properly certified HIBE ciphertext for epoch $t^{\star}$ (say $C^{\star} = C^{\star}_l$, for some $l \in \{1, \ldots, m\}$, where $C^{\star}_1, \ldots, C^{\star}_m$ are the HIBE ciphertexts of $RL_{t^{\star}}$). However, the execution of Open reveals a previously unseen $X^{\star}$ or points some revoked user $i \in U^a \cap \mathcal{R}_{t^{\star}}$ although $\sigma^{\star}$ provides convincing evidence that the committed private key $(D^{\star}_1, D^{\star}_2)$ allows decrypting $C^{\star}_l$ and that committed elements $\{\theta^{\star}_i\}_{i=1}^{7}$ form a valid signature on $(X^{\star}, D^{\star}_2)$. In this case, we have two situations:
  a. The pair $(X^{\star}, D^{\star}_2)$ was *not* signed by $\mathsf{J_{GM}}$ in any execution of Join. It means that either: (1) Open uncovers a value $X^{\star}$ that does not appear anywhere in St_{trans}. (2) The traced user $i \in U^a \cap \mathcal{R}_{t^{\star}}$ colluded with some unrevoked user $j \in U^a$ whose leaf *is* in S_{k_l,u_l} – which $C^{\star}_l$ is an encoding of – and managed to forge an AHO signature so as to link his $X^{\star}$ to an authorized key $(D^{\star}_1, D^{\star}_2)$ for S_{k_l,u_l}.
  b. The pair $(X^{\star}, D^{\star}_2)$ was signed by $\mathsf{J_{GM}}$ at some execution of Join. At first, we would like to use the Type II.b adversary to break the standard selective semantic security of the HIBE system (cf. Definition 7 in appendix B). As it turns out, even if we were using the original BBG HIBE (with its master secret key), such a reduction would be unlikely to work because the set S_{k_l,u_l} may contain unrevoked users in U^a, so that $\mathcal{A}$ obtained private keys that do decrypt $C^{\star}_l$. Instead, we rely on the security property that we call "key-robustness" (defined in appendix B.1) and which relies on a weaker assumption than the standard security of the BBG HIBE. Observe that, when user i joined the group, he cannot have been issued the private key $(D^{\star}_1, D^{\star}_2)$ (which is an authorized key for S_{k_l,u_l}) since he is revoked at epoch $t^{\star}$. However, since $(X^{\star}, D^{\star}_2)$ was signed by $\mathsf{J_{GM}}$, user i must have obtained from $\mathsf{J_{GM}}$ a HIBE private key of the form $(D_1, D^{\star}_2)$, where $D_1 \neq D^{\star}_1$, for an identity other than the one encoded by S_{k_l,u_l}. In this case, as will be showed in Lemma 2, the key-robustness property is necessarily broken in the HIBE scheme.

It is easy to see that Type I and Type II.a forgeries imply a forger against the AHO signature scheme (the proof is straightforward and omitted).

Lemma 2 (in appendix C) demonstrates that a Type II.b attack necessarily contradicts the key-robustness property (formally defined in appendix B.1) of the Boneh-Boyen-Goh HIBE scheme and thus the Diffie-Hellman assumption, as established by Lemma 1 in appendix B.2.

Finally, one can readily check that an adversary cannot produce a signature $\sigma^{\star}$ allowing to win the misidentification game without being one of the above kinds of forgeries. The result of the theorem follows from the fact that the SFP assumption implies the CDH assumption. □

The security against framing attacks and the anonymity property rely on the SDH and DLIN assumptions, respectively, and the proofs are given in appendices D.1 and D.2.

4 Efficiency Comparisons

This section discusses the comparative efficiency of known pairing-based revocable group signatures. We focus on revocation methods that are more efficient than generic revocation techniques: for example, we do not consider techniques (such as the one recalled in [19][Section 5.4]) consisting in privately sending new keys to all remaining users at each revocation. Also, we only consider schemes where group members are stateless and do not have to update their membership certificate every time a revocation occurs.

In table 1, all sizes are given in terms of number of group elements, each one of which costs $O(\lambda)$ bits to represent.

Table 1. Comparison between pairing-based revocable group signatures

Schemes	Group public key size	Signature size	Membership certificate size	Revocation list size	Signature cost	Verification cost	Revocation cost*	Standard model?
NFHF1 [49]	$O(N)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
NFHF2 [49]	$O(N^{1/2})$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(1)$	$O(r)$	✗
BS [18]	$O(1)$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(1)$	✗
NF [50]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✗
LV [48]	$O(T)^{\diamondsuit}$	$O(1)$	$O(1)$	$O(r)$	$O(1)$	$O(r)$	$O(r)$	✓
This work (SD)	$O(\log N)$	$O(1)$	$O(\log^3 N)$	$O(r)$	$O(\log N)^{\dagger\ \ddagger}$	$O(1)$	$O(r \cdot \log N)^{\ddagger}$	✓
This work (CS)	$O(1)$	$O(1)$	$O(\log N)$	$O(r \cdot \log(N/r))$	$O(1)^{\ddagger}$	$O(1)$	$O(r \cdot \log(N/r))^{\ddagger}$	✓

N: max. number of users; r: number of revocations T: max. number of revocation epochs

$*$ The revocation cost refers to the complexity of generating an up-to-date revocation list.
$\diamondsuit$ These schemes can be modified to have $O(1)$-size group public keys.
† This complexity is only involved at the first signature of each revocation epoch.
‡ We only count arithmetic operations. In the signing algorithm, for example, we neglect $O(\log \log N)$ combinatorial operations at the beginning of each epoch.

As we can see, our CS and SD-based constructions are not only the first revocable group signatures with constant verification time in the standard model. Among schemes where revocations do not entail updates in unrevoked users' credentials, they are also the only solutions offering $O(1)$ verification cost and at most poly-logarithmic complexity in other metrics.

In applications where one can afford a logarithmic expansion factor in the size of revocation lists, the CS method seems preferable as it features compact (meaning logarithmic according to the terminology used in [9, 19]) membership certificates.

In situations where the size of the revocation list is to be minimized, the SD method should be preferred. Alternatively, the Layered Subset Difference approach (LSD) [41] provides an interesting tradeoff: at the expense of doubling the maximal size of revocation lists (which asymptotically remain of size $O(r)$), its basic variant allows reducing the size of membership certificates to $O(\log^{5/2} N)$ as only $O(\log^{3/2} N)$ HIBE private keys have to be stored.

Acknowledgements

We thank Damien Vergnaud, Sébastien Canard, Roch Lescuyer and the anonymous reviewers for useful comments.

References

1. M. Abdalla, E. Kiltz, G. Neven. Generalized Key Delegation for Hierarchical Identity-Based Encryption. In *ESORICS'07*, *LNCS* 4734, pp. 139–154. Springer, 2007.
2. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
5. T. Acar, L. Nguyen. Revocation for Delegatable Anonymous Credentials. In *PKC'11*, *LNCS* 6571, pp. 423–440, 2011.
6. G. Ateniese, J. Camenisch, S. Hohenberger, B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive: Report 2005/385, 2005.
7. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto'00*, *LNCS* 1880, pp. 255–270, 2000.
8. G. Ateniese, D. Song, G. Tsudik. Quasi-Efficient Revocation in Group Signatures. In *Financial Cryptography'02*, *LNCS* 2357, pp. 183–197, 2002.

9. M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt'03*, *LNCS* 2656, pp. 614–629, 2003.
10. M. Bellare, P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM Press, 1993.
11. M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA'05*, *LNCS* 3376, pp. 136–153, 2005.
12. J. Benaloh, M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures. In *Eurocrypt'93*, *LNCS* 4948, pp. 274–285, 1993.
13. D. Boneh, X. Boyen. Short Signatures Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 56–73. Springer-Verlag, 2004.
14. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
15. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
16. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
17. D. Boneh, B. Lynn, H. Shacham. Short signatures from the Weil pairing. In *Asiacrypt'01*, *LNCS* 2248, pp. 514–532. Springer, 2001.
18. D. Boneh, H. Shacham. Group signatures with verifier-local revocation. In *ACM-CCS'04*, pp. 168–177. ACM Press, 2004.
19. X. Boyen, B. Waters. Compact Group Signatures Without Random Oracles. In *Eurocrypt'06*, *LNCS* 4004, pp. 427–444, Springer, 2006.
20. X. Boyen, B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *Crypto'06*, *LNCS* 4117, pp. 290–307, 2006.
21. X. Boyen, B. Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In *PKC'07*, *LNCS* 4450, pp. 1–15, 2007.
22. E. Bresson, J. Stern. Efficient Revocation in Group Signatures. In *PKC'01*, *LNCS* 1992, pp. 190–206, 2001.
23. E. Brickell. An efficient protocol for anonymously providing assurance of the container of the private key. Submission to the Trusted Computing Group. April, 2003.
24. E. Brickell, J. Camenisch, L. Chen. Direct Anonymous Attestation. In *ACM-CCS'04*, pp. 132–145, 2004.
25. J. Camenisch, R. Chaabouni, a. shelat. Efficient Protocols for Set Membership and Range Proofs. In *Asiacrypt'08*, *LNCS* 5350, pp. 234–252, Springer, 2008.
26. J. Camenisch, M. Dubovitskaya, G. Neven, G. Zaverucha. Oblivious Transfer with Hidden Access Control Policies. In *PKC'11*, *LNCS* 6571, pp. 192–209, 2011.
27. J. Camenisch, M. Kohlweiss, C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*, *LNCS* 5443, pp. 481–500, 2009.
28. J. Camenisch, M. Kohlweiss, C. Soriente. Solving Revocation with Efficient Update of Anonymous Credentials. In *SCN'10*, *LNCS* 6280, pp. 454–471, 2010.
29. J. Camenisch, A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto'02*, *LNCS* 2442, pp. 61–76, Springer, 2002.
30. R. Canetti, S. Halevi, J. Katz. A Forward-Secure Public-Key Encryption Scheme. In *Eurocrypt'03*, *LNCS* 2656, pp. 254–271, 2003.
31. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
32. D. Chaum, E. van Heyst. Group Signatures. In *Eurocrypt'91*, *LNCS* 547, pp. 257–265, Springer, 1991.
33. C. Delerablée, D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *Vietcrypt'06*, *LNCS* 4341, pp. 193–210, Springer, 2006.
34. Y. Dodis, N. Fazio. Public Key Broadcast Encryption for Stateless Receivers. In *Digital Rights Management (DRM'02)*, *LNCS* 2696, pp. 61–80, 2002.
35. C.-I. Fan, R.-H. Hsu, M. Manulis. Group Signature with Constant Revocation Costs for Signers and Verifiers. In *Cryptology and Network Security (CANS 2011)*, *LNCS* 7092, pp. 214–233, 2011.
36. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
37. C. Gentry, A. Silverberg. Hierarchical ID-based cryptography. In *Asiacrypt'02*, *LNCS* 2501, Springer, 2002.
38. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
39. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, *LNCS* 4833, pp. 164–180. Springer, 2007.
40. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
41. D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Crypto'02*, *LNCS* 2442, pp. 47–60, Springer, 2002.

42. D. Hofheinz, T. Jager, E. Kiltz. Short Signatures From Weaker Assumptions. In *Asiacrypt'11*, *LNCS* series, to appear, 2011.
43. D. Hofheinz, E. Kiltz. Programmable hash functions and their applications. In *Crypto'08*, *LNCS* 5157, pp. 21–38, 2008.
44. J. Horwitz, B. Lynn. Toward hierarchical identity-based encryption. In *Eurocrypt'02*, *LNCS* 2332, Springer, 2002.
45. A. Kiayias, M. Yung. Secure scalable group signature with dynamic joins and separable authorities. International Journal of Security and Networks (IJSN) Vol. 1, No. 1/2, pp. 24–45, 2006. Earlier version appeared as Cryptology ePrint Archive: Report 2004/076, 2004.
46. A. Kiayias, M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt'05*, *LNCS* 3494, pp. 198–214, 2005.
47. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, *LNCS* 3876, pp. 581–600, 2006.
48. B. Libert, D. Vergnaud. Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model. In *CANS'09*, *LNCS* 5888, pp. 498-517, 2009.
49. T. Nakanishi, H. Fujii, Y. Hira, N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC'09*, *LNCS* 5443, pp. 463–480, 2009.
50. T. Nakanishi, N. Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *Asiacrypt'05*, *LNCS* 5443, pp. 533–548, 2009.
51. M. Naor, D. Naor, J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. In *Crypto'01*, *LNCS* 2139, pp. 41–62, 2001.
52. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer-Verlag, 2003.
53. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA'05*, *LNCS* 3376, pp. 275–292, 2005.
54. L. Nguyen, R. Safavi-Naini. Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings. In *Asiacrypt'04*, *LNCS* 3329, pp. 372–386. Springer-Verlag, 2004.
55. D. Song. Practical forward secure group signature schemes. In *ACM-CCS'01*, pp. 225–234, 2001.
56. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt'97*, *LNCS* 1233, pp. 256–66, 1997.
57. I. Teranishi, K. Sako. k-Times Anonymous Authentication with a Constant Proving Cost. In *PKC'06*, *LNCS* 3958, pp. 525–542, Springer, 2006.
58. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. Blacklistable anonymous credentials: blocking misbehaving users without TTPs. In *ACM-CCS'07*, pp. 72–81, 2007.
59. P. Tsang, M.-Ho Au, A. Kapadia, S. Smith. PEREA: towards practical TTP-free revocation in anonymous authentication. In *ACM-CCS'08*, pp. 333–344, 2008.
60. G. Tsudik, S. Xu. Accumulating Composites and Improved Group Signing. In *Asiacrypt'03*, *LNCS* 2894, pp. 269–286, 2003.
61. S. Zhou, D. Lin. Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In *CANS'06*, *LNCS* 4301, pp. 126–143, Springer, 2006.

A Correctness and Security Definitions for Revocable Group Signatures

In the following, a public state St is said *valid* if it can be reached from $St = (\emptyset, \varepsilon)$ by a Turing machine having oracle access to J_{GM}. Likewise, a state St' is said to *extend* another state St if it can be reached from St.

Similarly to [45, 46], we will write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$ to express that there exist coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state St', the execution of $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle i, \mathsf{sec}_i, \mathsf{cert}_i \rangle$.

Correctness. We say that a R-GS scheme is correct if:

1. In a valid state St, it holds that $|St_{users}| = |St_{trans}|$ and no two entries of St_{trans} can contain certificates with the same tag.
2. If $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$ is honestly run by both parties and $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then it holds that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.
3. For each revocation epoch t and any $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, if $i \notin \mathcal{R}_t$, it holds that $\mathsf{Verify}\big(\mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M), M, t, RL_t, \mathcal{Y}\big) = 1$.
4. For any $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ resulting from the interaction $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$ for some valid state St, any revocation epoch t such that $i \notin \mathcal{R}_t$, if $\sigma = \mathsf{Sign}(\mathcal{Y}, t, RL_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then

$$\mathsf{Open}(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') = i.$$

Security Model. As in [45], we formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes oracles. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$. It comprises the (initially empty) set St_{users} of group members and a database St_{trans} containing transcripts of join protocols. Finally, $\mathsf{state}_{\mathcal{I}}$ includes a counter t (which is initialized to 0) indicating the number of user revocation queries so far.
- $n = |St_{users}| < N$ is the current cardinality of the group.
- Sigs: is a database of signatures issued by the signing oracle. Each record is a triple (i, t, M, σ) indicating that message M was signed by user i during period t.
- U^a: is the set of users that are adversarially-controlled since their introduction in the system.
- U^b: is the set of honest users that were introduced by the adversary acting as a dishonest group manager. For such users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

When mounting attacks, adversaries will be granted access to the following oracles.

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under his control in the group. On behalf of the GM, the interface interacts with the malicious prospective user by running J_{GM} in the join protocol. If the protocol successfully terminates, the interface increments N, updates St by inserting the new user n in sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a dishonest group manager, to introduce new group members of his choice. The interface starts an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the J_{GM}-executing adversary. If the protocol successfully completes, the interface increments n, adds user n to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks if the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If no such elements exist or if $i \notin U^b$, it returns $\perp$. Otherwise, it generates a signature σ on behalf of user i for epoch t. It also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs} || (i, t, M, \sigma)$.
- Q_{open}: on input of a valid pair (M, σ) for some revocation epoch t, the interface runs the opening algorithm using the current state St. When S is a set of triples (M, σ, t), $Q_{\mathsf{open}}^{\neg S}$ denotes the restricted oracle that applies the opening procedure to any triple (M, σ, t) but those in S.
- Q_{read} and Q_{write}: allow the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. When invoked, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. Queries Q_{write} allow the adversary to modify $\mathsf{state}_{\mathcal{I}}$ as long as it does not remove or alter elements of St_{users}, St_{trans} or invalidate the public state St: for example, the adversary can use it to create dummy users at will as long as it does not re-use already existing certificate tags.
- Q_{revoke}: is a user revocation oracle. On input of an index i such that $i \in St_{users}$, the interface checks if i is in the appropriate user set (*i.e.*, U^a or U^b depending on the considered security notion) and if St_{trans} contains a record $\langle i, \mathsf{transcript}_i \rangle$ such that $i \notin \mathcal{R}_t$, where t is the current revocation epoch. If not, it returns $\perp$. Otherwise, it increments t, adds i to $\mathcal{R}_t$ and generates an updated revocation list RL_t which is made available to the adversary. For simplicity, we assumed that the adversary only revokes one user per query to Q_{revoke} but the model easily extends to allow multiple revocations at once.

The KY model considers properties termed security against *misidentification attacks*, *framing attacks* and *anonymity*.

In a misidentification attack, the adversary is allowed to corrupt the opening authority via the Q_{keyOA} oracle. He can also introduce corrupt users in the group via $Q_{\mathsf{a\text{-}join}}$-queries and revoke users at will using Q_{revoke}. His goal is to produce a signature $\sigma^\star$ that verifies w.r.t. $RL_{t^\star}$, where $t^\star$ denotes the current revocation epoch (*i.e.*, the number of Q_{revoke}-queries). It wins if the produced signature $\sigma^\star$ does not open to any unrevoked adversarially-controlled.

Definition 4. *A R-GS scheme is secure against misidentification attacks if, for any PPT adversary $\mathcal{A}$ involved in the experiment hereafter, we have* $\mathbf{Adv}^{\text{mis-id}}_{\mathcal{A}}(\lambda) = \Pr[\mathbf{Expt}^{\text{mis-id}}_{\mathcal{A}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Expt}^{\text{mis-id}}_{\mathcal{A}}(\lambda)$
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$;
If $\mathsf{Verify}(\sigma^\star, M, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ *return* 0;
$i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
If $(i \notin U^a \backslash \mathcal{R}_{t^\star})$ *return* 1;
Return 0;

This definition extends the usual definition [45] in that $\mathcal{A}$ is also successful if $\sigma^\star$ verifies w.r.t. $RL_{t^\star}$ but opens to an adversarially-controlled user that was revoked during the revocation epoch $t^\star$.

Framing attacks consider the situation where the whole system, including the group manager and the opening authority, conspires against some honest user. The adversary is allowed to corrupt the group manager *and* the opening authority (using Q_{keyGM} and Q_{keyOA}, respectively). He can also introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users generate signatures and create dummy users using Q_{write}. In addition, before the possible corruption of the group manager, the adversary can revoke group members at any time by invoking the Q_{revoke} oracle. As a potentially corrupted group manager, $\mathcal{A}$ is allowed to come up with his own revocation list $RL_{t^\star}$ at the end of the game. We assume that anyone can publicly verify that $RL_{t^\star}$ is correctly formed (*i.e.*, that it could be a legitimate output of Revoke) so that the adversary does not come up with an ill-formed revocation list. For consistency, if $\mathcal{A}$ chooses not to corrupt the GM, the produced revocation list $RL_{t^\star}$ must be the one determined by the history of Q_{revoke}-queries. The adversary eventually aims at framing an uncorrupt group member.

Definition 5. *A R-GS scheme is secure against framing attacks if, for any PPT adversary $\mathcal{A}$, it holds that* $\mathbf{Adv}^{\text{fra}}_{\mathcal{A}}(\lambda) = \Pr[\mathbf{Expt}^{\text{fra}}_{\mathcal{A}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Expt}^{\text{fra}}_{\mathcal{A}}(\lambda)$
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
$(M^\star, \sigma^\star, t^\star, RL_{t^\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{revoke}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
If $\mathsf{Verify}(\sigma^\star, M^\star, t^\star, RL_{t^\star}, \mathcal{Y}) = 0$ *then return* 0;
$i = \mathsf{Open}(M^\star, t^\star, RL_{t^\star}, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
If $i \notin U^b$ *return* 0;
If $\big(\bigwedge_{j \in U^b \ s.t.\ j=i} (j, t^\star, M^\star, *) \notin \mathsf{Sigs}\big)$ *then return* 1;
Return 0;

Anonymity is defined via a game involving a 2-stage adversary. In the first stage, called play stage, the adversary is allowed to modify $\mathsf{state}_{\mathcal{I}}$ by making Q_{write}-queries and to open signatures of his choice by invoking Q_{open}. At the end of the play stage, it chooses a message-period pair $(M^\star, t^\star)$ and two pairs $(\mathsf{sec}^\star_0, \mathsf{cert}^\star_0)$, $(\mathsf{sec}^\star_1, \mathsf{cert}^\star_1)$, consisting of a well-formed membership certificate and a membership secret for $b = 0, 1$. The challenger flips a fair binary coin $d \xleftarrow{R} \{0, 1\}$ and generates a signature $\sigma^\star$ using $(\mathsf{sec}^\star_d, \mathsf{cert}^\star_d)$. The adversary aims to eventually determine the bit d. Of course, it is restricted not to query the opening of $(M^\star, \sigma^\star)$ during the guess stage.

Definition 6. *A R-GS scheme is fully anonymous if* $\mathbf{Adv}^{\text{anon}}(\mathcal{A}) := |\Pr[\mathbf{Expt}^{\text{anon}}_{\mathcal{A}}(\lambda) = 1] - 1/2|$ *is negligible for any PPT adversary* $\mathcal{A}$ *involved in the following experiment:*

Experiment $\mathbf{Expt}^{\text{anon}}_{\mathcal{A}}(\lambda)$
$\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda)$;
$\big(aux, M^\star, t^\star, RL_{t^\star}, (\mathsf{sec}^\star_0, \mathsf{cert}^\star_0), (\mathsf{sec}^\star_1, \mathsf{cert}^\star_1)\big)$
$\quad \leftarrow \mathcal{A}(\mathsf{play} : Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{revoke}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
If $\neg(\mathsf{cert}_b \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b)$ *for* $b \in \{0,1\}$ *or if* $\mathsf{cert}_0 = \mathsf{cert}_1$ *return* 0;
$d \xleftarrow{R} \{0,1\}$; $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, \mathsf{cert}^\star_d, \mathsf{sec}^\star_d, M^\star)$;
$d' \leftarrow \mathcal{A}(\mathsf{guess} : \sigma^\star, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star,\sigma^\star,t^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
If $d' = d$ *then return* 1;
Return 0;

B Hierarchical Identity-Based Encryption

Consider a hierarchy of entities, each of which has a unique address $\mathsf{ID} = (I_1, \ldots, I_\ell)$, with $I_i \in \{0,1\}^*$ for $1 \leq i \leq \ell$, at level ℓ. For any $i \leq \ell$, $\mathsf{ID}|_i$ denotes the prefix $(I_1, \ldots, I_i)$ of ID. The address of a node at level i is obtained by appending its local identifier I_i to its father's address $\mathsf{ID}|_{i-1}$.

A HIBE scheme [44, 37] is a tuple $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Derive}, \mathsf{Encrypt}, \mathsf{Decrypt})$ of algorithms[5] working as follows. Setup is run by a trusted private key generator (PKG) to generate a master public key mpk and a master secret key msk. The latter is used by the PKG, at the root of the hierarchy, to derive private keys from users' identities at level 1. The key generation algorithm Keygen takes as input the master secret key msk and a hierarchical identity $\mathsf{ID} = (I_1, \ldots, I_k)$ and returns a private key d_{ID} for that identity. Algorithm Derive is used by a ℓ-th level entity with address $\mathsf{ID} = (I_1, \ldots, I_\ell)$ to compute private keys for its children labeled as $(I_1, \ldots, I_\ell, *)$ at depth $\ell + 1$. It takes in a ℓ-th level private key d_{ID} and a vector $\mathsf{ID}' = (I_1, \ldots, I_\ell, I_{\ell+1})$, where $\mathsf{ID}'|_\ell = \mathsf{ID}$, to generate a $(\ell+1)$-th level secret key $d_{\mathsf{ID}'}$. Algorithm $\mathsf{Encrypt}$ takes in a plaintext $m \in \mathcal{M}$, where $\mathcal{M}$ denotes the plaintext space, the master public key mpk and the receiver's address $\mathsf{ID} = (I_1, \ldots, I_\ell)$ to produce a ciphertext C that can be undone by the receiver having obtained d_{ID} from its father.

In many HIBE constructions (such as [15, 20]) delegated private keys (produced by Derive) have the same distribution as original keys (generated by Keygen). In the upcoming security definitions, we assume that this property is satisfied by the considered HIBE system.

In the following, we say that a HIBE scheme is *key-partitioned* if private keys $d_{\mathsf{ID}} = (D_{\mathsf{ID}}, K_{\mathsf{ID}})$ consist of two distinct part: the first one D_{ID} is called *decryption component* and it is only used to decrypt messages; the second part K_{ID} is called *delegation component* and its sole purpose is to derive private keys for children nodes. Many HIBE systems in the literature (e.g., [15, 20]) are key-partitioned.

B.1 Security Definitions for HIBE

The standard security notion [37] captures that any coalition of hierarchy entities that are not ancestors of some user should be unable to gain information on messages encrypted for that user.

In [30], Canetti, Halevi and Katz suggested a weaker security notion, called *selective* security, where the adversary has to choose its target identity upfront.

Definition 7. *[30] A HIBE system with* ℓ *levels is selectively secure (or* IND-sID-CPA *secure) if no PPT adversary* $\mathcal{A}$ *has non-negligible advantage in this game:*

[5] We use the syntax of [20] which involves an explicit delegation algorithm Derive. Although this algorithm is not explicitly written in [15], it exists as noted in appendix B.2.

1. *The adversary $\mathcal{A}$ chooses a target identity $\mathsf{ID}^\star = (I_1^\star, \ldots, I_{\ell^\star}^\star)$ at depth $\ell^\star < \ell$, for some $\ell^\star$ of its choice. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$ and hands mpk to $\mathcal{A}$.*
2. *$\mathcal{A}$ issues a number of key extraction queries under the rule that no prefix of $\mathsf{ID}^\star$ can be the input of a key extraction query. On input of an identity $\mathsf{ID} = (I_1, \ldots, I_k)$, with $k \leq \ell$, the challenger responds with $d_{\mathsf{ID}} \leftarrow \mathsf{Keygen}(msk, \mathsf{ID})$.*
3. *When $\mathcal{A}$ decides that the first phase is over, it chooses messages m_0, m_1. The challenger flips a coin $d \xleftarrow{R} \{0,1\}$ and responds with a challenge $C^\star = \mathsf{Encrypt}(mpk, \mathsf{ID}^\star, m_d)$.*
4. *$\mathcal{A}$ issues new queries but cannot ask for the private key of a prefix of $\mathsf{ID}^\star$.*
5. *$\mathcal{A}$ finally outputs a bit $d' \in \{0,1\}$ and wins if $d' = d$. As usual, $\mathcal{A}$'s advantage is quantified as the distance $\mathbf{Adv}^{\mathrm{hibe}}(A) := |\Pr[d' = d] - 1/2|$.*

For our purposes, we need a different and non-standard form of selective security, which mandates that the adversary be unable to maul a private key that it obtained for some target identity $\mathsf{ID}^\dagger$ *even* knowing the master secret key. By "mauling", we mean computing a private key for a different identity $\mathsf{ID}' \neq \mathsf{ID}^\dagger$ but under the same randomness as the received key $d_{\mathsf{ID}^\dagger}$.

Definition 8. *A key-partitioned HIBE system with ℓ levels is said selectively key-robust if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the following game:*

1. *The adversary $\mathcal{A}$ chooses an identity $\mathsf{ID}^\dagger = (I_1^\dagger, \ldots, I_{\ell^\dagger}^\dagger)$ that it wishes to be challenged upon at the depth $\ell^\dagger < \ell$ of its choice. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(\lambda)$ and hands (msk, mpk) to $\mathcal{A}$ along with a challenge consisting of a private key $d_{\mathsf{ID}^\dagger}$ for the identity $\mathsf{ID}^\dagger$.*
2. *$\mathcal{A}$ outputs an identity ID' such that $\mathsf{ID}^\dagger$ is not a prefix[6] of ID' and a decryption component $D_{\mathsf{ID}'}$ (i.e., a private key without delegation component). The adversary wins if: (i) $D_{\mathsf{ID}'}$ is a valid decryption component for ID'; (ii) $D_{\mathsf{ID}'}$ and $d_{\mathsf{ID}^\dagger}$ correspond to the same randomness of the key generation algorithm.*

In Definition 8, we insist that $\mathcal{A}$ is given a full private key for the target identity $\mathsf{ID}^\dagger$ but it only has to output a valid decryption component $D_{\mathsf{ID}'}$ for ID'.

It is also worth insisting that, for the application of this paper, a selective flavor of key-robustness suffices. Indeed, since the number of group members is always polynomial, the target identity $\mathsf{ID}^\dagger$ can be guessed upfront with non-negligible probability in the proof of Lemma 2 (in appendix C).

B.2 The Boneh-Boyen-Goh HIBE

In [15], Boneh, Boyen and Goh (BBG) described the first HIBE scheme where the size of ciphertext does not depend on the depth of the receiver in the hierarchy. The construction bears resemblance with the first selectively secure IBE scheme of Boneh and Boyen [14], which can be seen as a single-level variant of the BBG HIBE. The latter works as follows.

$\mathsf{Setup}(\lambda, \ell)$: given a security parameter $\lambda \in \mathbb{N}$ and the number of levels $\ell \in \mathbb{N}$ in the hierarchy, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p, where $p > 2^\lambda$. Choose $\alpha \xleftarrow{R} \mathbb{Z}_p$, $g, g_2, h_0, h_1, \ldots, h_\ell \xleftarrow{R} \mathbb{G}$ and compute $g_1 = g^\alpha$. The master public key is defined to be

$$mpk_{\mathsf{BBG}} := \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g_1,\ g_2,\ \{h_i\}_{i=0}^{\ell}\Big)$$

while the master secret key consists of $msk_{\mathsf{BBG}} := g_2^\alpha$. The space of hierarchical identities is $\mathcal{I} = (\mathbb{Z}_p^*)^{\leq \ell}$.

[6] We assume that hierarchical identities are prefixes of themselves for simplicity.

Keygen$\big(msk_{\mathsf{BBG}}, \mathsf{ID} = (I_1, \ldots, I_k)\big)$**:** to generate a private key for $\mathsf{ID} = (I_1, \ldots, I_k) \in (\mathbb{Z}_p^*)^k$ at level k using $\mathsf{msk}_{\mathsf{BBG}} = g_2^\alpha$, choose $r \xleftarrow{R} \mathbb{Z}_p$. Then, compute and return

$$d_{\mathsf{ID}} = (D_1, D_2, K_{k+1}, \ldots, K_\ell) = \Big(g_2^\alpha \cdot \big(h_0 \cdot \prod_{i=1}^{k} h_i^{I_i}\big)^r,\ g^r,\ h_{k+1}^r, \ldots,\ h_\ell^r\Big) \in \mathbb{G}^{\ell-k+2}. \tag{10}$$

Derive$\big(d_{\mathsf{ID}}, \mathsf{ID}' = (I_1, \ldots, I_k, I_{k+1})\big)$**:** given a private key d_{ID} of the form (10) for the hierarchical identity $\mathsf{ID} = (I_1, \ldots, I_k)$, it is easy to derive a key for the identity $\mathsf{ID}' = (I_1, \ldots, I_k, I_{k+1}) \in (\mathbb{Z}_p^*)^{k+1}$ by choosing $r' \xleftarrow{R} \mathbb{Z}_p$ and computing

$$\begin{aligned} d_{\mathsf{ID}'} &= (D_1', D_2', K_{k+2}', \ldots, K_\ell') \\ &= \Big(D_1 \cdot K_{d+1}^{I_{k+1}} \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^{r'}, D_2 \cdot g^{r'}, K_{k+2} \cdot h_{k+2}^{r'}, \ldots, K_\ell \cdot h_\ell^{r'}\Big) \\ &= \Big(g_2^\alpha \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^{r''},\ g^{r''},\ h_{k+2}^{r''}, \ldots,\ h_\ell^{r''}\Big) \in \mathbb{G}^{\ell-k+1}, \end{aligned} \tag{11}$$

where $r'' = r + r'$.

Encrypt$(mpk_{\mathsf{BBG}}, \mathsf{ID} = (I_1, \ldots, I_d), M)$**:** to encrypt $M \in \mathbb{G}$ under $\mathsf{ID} = (I_1, \ldots, I_d) \in (\mathbb{Z}_p^*)^d$, choose $s \xleftarrow{R} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot e(g_1, g_2)^s, \qquad C_1 = g^s, \qquad C_2 = \big(h_0 \cdot h_1^{I_1} \cdots h_d^{I_d}\big)^s$$

The ciphertext is $C = \big(C_0, C_1, C_2\big)$.

Decrypt$(mpk_{\mathsf{BBG}}, d_{\mathsf{ID}}, C)$**:** parse d_{ID} as $(D_1, D_2, K_{d+1}, \ldots, K_\ell) \in \mathbb{G}^{\ell-d+2}$ and the ciphertext C as $\big(C_0, C_1, C_2\big)$. Then, compute and output

$$M = C_0 \cdot e(C_1, D_1)^{-1} \cdot e(C_2, D_2).$$

It is easy to see that this construction is key-partitioned since the private key can be divided into $(D_{\mathsf{ID}}, K_{\mathsf{ID}})$, where $D_{\mathsf{ID}} = (D_1, D_2) \in \mathbb{G}^2$ is only used to decrypt and $K_{\mathsf{ID}} = (K_{k+1}, \ldots, K_\ell) \in \mathbb{G}^{\ell-k+2}$ is only useful for delegations.

When the private key for ID' is derived from the private key for ID, the randomizer $r' \in \mathbb{Z}_p$ in (11) allows making sure that derived private keys are indistinguishable from original keys that are generated directly at level $k+1$.

In our application, we will require that key be derived without any randomization. Namely, a private key for ID' is always derived as per

$$\begin{aligned} d_{\mathsf{ID}'} = (D_1', D_2', K_{k+2}', \ldots, K_\ell') &= \Big(D_1 \cdot K_{d+1}^{I_{k+1}},\ D_2,\ K_{k+2}, \ldots,\ K_\ell\Big) \\ &= \Big(g_2^\alpha \cdot \big(h_0 \cdot \prod_{i=1}^{k+1} h_i^{I_i}\big)^r,\ g^r,\ h_{k+2}^r, \ldots,\ h_\ell^r\Big) \in \mathbb{G}^{\ell-k+1}. \end{aligned}$$

For this reason, a private key and its descendants will always share the same component D_2. However, it does not affect the security of the group signature since, in the join protocol, users are always given freshly generated HIBE private keys.

The following lemma demonstrates that the BBG HIBE is selectively key-robust under the Diffie-Hellman assumption. The proof implicitly relies on the fact (implicitly noted in [43, 42]) that BLS-type signatures [17] can be proved secure in the standard model when the number of signing queries is bounded by a small constant (such as one here since the one-time public key g^r is used as a one-time public key).

Lemma 1. *The BBG HIBE scheme is selectively key-robust assuming that the CDH assumption holds in $\mathbb{G}$. More precisely, a selective key-robustness adversary $\mathcal{A}$ with advantage ε implies an algorithm $\mathcal{B}$ solving the CDH problem with advantage $\varepsilon \cdot (1 - 1/p)$.*

Proof. Towards a contradiction, let us assume that a selective adversary $\mathcal{A}$ has non-negligible advantage in the game of Definition 8. We show that $\mathcal{A}$ allows breaking the CDH assumption.

Algorithm $\mathcal{B}$ receives as input a CDH instance $(g, g^a, g^b) \in \mathbb{G}^3$ and undertakes to compute g^{ab}. At the beginning of its interaction with $\mathcal{A}$, the latter chooses a target identity $\mathsf{ID}^\dagger = (I_1^\dagger, \ldots, I_{\ell^\dagger}^\dagger) \in (\mathbb{Z}_p^*)^{\ell^\dagger}$ at the depth $\ell^\dagger \leq \ell$ of its choice. Then, $\mathcal{B}$ generates the master key pair $(msk_{\mathsf{BBG}}, mpk_{\mathsf{BBG}})$ by choosing $\alpha \xleftarrow{R} \mathbb{Z}_p$, $g_2 \xleftarrow{R} \mathbb{G}$ and setting $g_1 = g^\alpha$ as in the normal setup algorithm. Then, it picks $\gamma_1, \ldots, \gamma_{\ell^\dagger} \xleftarrow{R} \mathbb{Z}_p$, $\delta_0, \ldots, \delta_\ell \xleftarrow{R} \mathbb{Z}_p$ and defines

$$\begin{aligned} h_0 &= g^{\delta_0} \cdot (g^b)^{-\sum_{i=1}^{\ell^\dagger} \gamma_i I_i^\dagger} \\ h_i &= g^{\delta_i} \cdot (g^b)^{\gamma_i} && \text{for } i = 1, \ldots, \ell^\dagger \\ h_i &= g^{\delta_i} && \text{for } i = \ell^\dagger + 1, \ldots, \ell. \end{aligned}$$

To generate a private key $d_{\mathsf{ID}^\dagger} = (D_1, D_2, K_{\ell^\dagger+1}, \ldots, K_\ell)$ for the target identity $\mathsf{ID}^\dagger$, $\mathcal{B}$ sets

$$\begin{aligned} D_1 &= g_2^\alpha \cdot (g^a)^{\delta_0 + \sum_{i=1}^{\ell^\dagger} \delta_i I_i^\dagger} \\ D_2 &= g^a \\ K_i &= (g^a)^{\delta_i} && \text{for } i = \ell^\dagger + 1, \ldots, \ell, \end{aligned}$$

which form a valid private key for the random exponent $r = a$.

The adversary $\mathcal{A}$ is given $(mpk_{\mathsf{BBG}}, msk_{\mathsf{BBG}} = g_2^\alpha)$ and the private key $d_{\mathsf{ID}^\dagger}$. Its goal will be to produce a valid decryption component $D_{\mathsf{ID}'} = (D_1', D_2')$ corresponding to an identity $\mathsf{ID}' = (I_1', \ldots, I_k') \in (\mathbb{Z}_p^*)^k$, for some $k \in \{1, \ldots, \ell\}$, that $\mathsf{ID}^\dagger$ is not a prefix of. In addition, $D_{\mathsf{ID}'}$ should correspond to the same random exponent $r = a$ as $d_{\mathsf{ID}^\dagger}$ (in other words, $D_1' = D_2 = g^a$).

When $\mathcal{A}$ outputs its result $(\mathsf{ID}', D_{\mathsf{ID}'})$, we distinguish the following situations.

- If $k \leq \ell^\dagger$, we have

$$h_0 \cdot \prod_{i=1}^{k} h_i^{I_i'} = g^{\delta_0 + \sum_{i=1}^{k} \delta_i I_i'} \cdot (g^b)^{\sum_{i=1}^{k} \gamma_i \cdot (I_i' - I_i^\dagger) - \sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger} \tag{12}$$

  and, with overwhelming probability $1 - 1/p$, it holds that

$$\sum_{i=1}^{k} \gamma_i \cdot (I_i' - I_i^\dagger) - \sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger \neq 0. \tag{13}$$

  Indeed, the vector $\vec{\gamma} = (\gamma_1, \ldots, \gamma_{\ell^\dagger})$ is chosen uniformly in $\mathbb{Z}_p^{\ell^\dagger}$ and it is independent of $\mathcal{A}$'s view.
  - If $k = \ell^\dagger$, we must have $I_i' \neq I_i^\dagger$ for at least one $i \in \{1, \ldots, \ell^\dagger\}$. Since the coordinates of $\vec{\gamma}$ are independent and uniformly distributed, the probability to have $\sum_{i=1}^{\ell^\dagger} \gamma_i \cdot (I_i' - I_i^\dagger) = 0$ is at most $1/p$ since we are bounding the probability of a random vector $\vec{\gamma}$ to be orthogonal to a given non-zero vector of $\mathbb{Z}_p^{\ell^\dagger}$.
  - If $k < \ell^\dagger$, we may have $I_i' = I_i^\dagger$ for each $i \in \{1, \ldots, k\}$ (*i.e.*, ID' may be a prefix of $\mathsf{ID}^\dagger$). In this situation, the probability to have $\sum_{i=k+1}^{\ell^\dagger} \gamma_i I_i^\dagger = 0$ is also $1/p$ since $(\gamma_{k+1}, \ldots, \gamma_{\ell^\dagger})$ is independent of $\mathcal{A}$'s view and identities $I_{k+1}^\dagger, \ldots, I_{\ell^\dagger}^\dagger$ are always non-zero. Finally, if ID' is not a prefix of $\mathsf{ID}^\dagger$, there exists $i \in \{1, \ldots, k\}$ such that $I_i' \neq I_i^\dagger$. Then, the same argument as in previous cases applies.

Since $\mathcal{A}$ presumably outputs a decryption component $D_{\mathsf{ID}'} = (D_1, D_2) = \big(g_2^{\alpha} \cdot (h_0 \cdot \prod_{i=1}^{k} h_i^{I'_i})^a,\ g^a\big)$ with non-negligible probability ε, $\mathcal{B}$ can compute

$$g^{ab} = \Big(\frac{D_1}{g_2^{\alpha} \cdot (g^a)^{\delta_0 + \sum_{i=1}^{k} \delta_i I'_i}}\Big)^{1/(\sum_{i=1}^{k} \gamma_i \cdot (I'_i - I_i^{\dagger}) - \sum_{i=k+1}^{\ell^{\dagger}} \gamma_i I_i^{\dagger})}$$

with probability $\varepsilon \cdot (1 - 1/p)$.

- If $k > \ell^{\dagger}$, there exists $i \in \{1, \ldots, \ell^{\dagger}\}$ such that $I'_i \neq I_i^{\dagger}$ since $\mathsf{ID}^{\dagger}$ cannot be a prefix of ID'. In this case, we can write

$$h_0 \cdot \prod_{i=1}^{k} h_i^{I'_i} = g^{\delta_0 + \sum_{i=1}^{k} \delta_i I'_i} \cdot (g^b)^{\sum_{i=1}^{\ell^{\dagger}} \gamma_i \cdot (I'_i - I_i^{\dagger})}$$

where $\sum_{i=1}^{\ell^{\dagger}} \gamma_i \cdot (I'_i - I_i^{\dagger}) \neq 0$ with probability at least $1 - 1/p$. Then, the CDH solution g^{ab} can be found in the same way as in the case $k \leq \ell^{\dagger}$.

□

C Deferred Lemma for the Security against Misidentification Attacks

Lemma 2. *The advantage of any Type II.b forger $\mathcal{A}$ is at most*

$$\mathbf{Adv}_{\mathcal{A}}^{\text{mis-id-II.b}}(\lambda) \leq 4 \cdot N^2 \cdot \Big(1 - \frac{1}{p}\Big) \cdot \mathbf{Adv}^{\text{CDH}}(\lambda)$$

where N denotes the maximal number of users.

Proof. At the beginning of its interaction with its challenger, our selective key-robustness adversary $\mathcal{B}$ chooses a random node $x_j \in \mathsf{T}$ and a random descendant x'_j of x_j (alternatively, $\mathcal{B}$ can more simply choose two distinct random nodes in the tree and, with some probability, x'_j will be in the subtree rooted at x_j). Since x'_j is a descendant of x_j, its label $\langle x'_j \rangle$ can be written $\langle x'_j \rangle = \langle x_j \rangle || w_{\ell_1} \ldots w_{\ell_2}$, for some integers $\ell_1, \ell_2 \in \{1, \ldots, \ell\}$ and where $w_i \in \{0,1\}$ for each $i \in \{\ell_1, \ldots, \ell_2\}$. Then, $\mathcal{B}$ declares $\mathsf{ID}^{\dagger} = (\mathcal{H}(\langle x_j \rangle), \mathcal{H}(w_{\ell_1}), \ldots, \mathcal{H}(w_{\ell_2}))$ as its target identity at level $\ell_2 - \ell_1 + 2$. The key-robustness challenger replies by returning a master key pair $(msk_{\mathsf{BBG}}, mpk_{\mathsf{BBG}})$ consisting of

$$mpk_{\mathsf{BBG}} = \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g_1 = g^{\alpha},\ g_2,\ \{h_i\}_{i=0}^{\ell}\Big), \qquad msk_{\mathsf{BBG}} = g_2^{\alpha}$$

together with a private key $d_{\mathsf{ID}^{\dagger}} = (D_1^{\dagger}, D_2^{\dagger}, K_{\ell_2 - \ell_1 + 3}^{\dagger}, \ldots, K_{\ell}^{\dagger})$ for the identity $\mathsf{ID}^{\dagger}$.

Then, $\mathcal{B}$ uses mpk_{BBG} to construct the group public key $\mathcal{Y}$ and generates all other public key elements (including $pk_{\mathsf{AHO}}^{(0)}$ and $pk_{\mathsf{AHO}}^{(1)}$) according to the specification of the setup algorithm. In particular, $\mathcal{B}$ retains the group manager's secret key $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and uses it to answer $Q_{\mathsf{a\text{-}join}}$-queries.

At each $Q_{\mathsf{a\text{-}join}}$-query, $\mathcal{B}$ executes the join protocol on behalf of J_{GM} and proceeds exactly as the real J_{GM} does (recall that it knows $\mathcal{S}_{\mathsf{GM}}$ and can thus perfectly simulate J_{GM}) with one exception. Namely, when executing step b.1 of Join, if the private key D_w has to be computed for the target identity $\mathsf{ID}^{\dagger}$, $\mathcal{B}$ uses the private key $d_{\mathsf{ID}^{\dagger}}$ that it received from its challenger to compute

$$(D_{w,1}, D_{w,2}, K_{w,\ell_2 - \ell_1 + 3}, \ldots, K_{w,\ell}) = \big(D_1^{\dagger}/g_2^{\alpha},\ D_2^{\dagger},\ K_{\ell_2 - \ell_1 + 3}^{\dagger}, \ldots,\ K_{\ell}^{\dagger}\big).$$

As for Q_{pub}, Q_{revoke}, Q_{read} and Q_{OA}-queries, $\mathcal{B}$ simply answers them as the real oracles would.

If the game terminates and $\mathcal{B}$ did not have to compute a private key for $\mathsf{ID}^{\dagger}$ at some $Q_{\mathsf{a\text{-}join}}$-query, then

$\mathcal{B}$ halts and reports failure since it must have guessed the wrong tree nodes x_j and x'_j. Otherwise (*i.e.*, if $d_{\mathsf{ID}^\dagger}$ was used to compute part of a membership certificate), we know that $\mathcal{A}$'s forgery $\sigma^\star$ will contain a committed HIBE private key $(D_1^\star, D_2^\star)$ and a committed value $X^\star$ such that $(X^\star, D_2^\star)$ is one of the pairs that were signed by $\mathcal{B}$ during some $Q_{\mathsf{a\text{-}join}}$-query. Moreover, the pair $(X^\star, D_2^\star)$ was associated with some HIBE private key $d_{\mathsf{ID}^\diamond} = (D_1, D_2^\star, K_{\ell_2-\ell_1+3}, \ldots, K_\ell)$ for a certain hierarchical identity $\mathsf{ID}^\diamond$ at step b.1 of Join. Since that identity is entirely defined by two nodes at the extremities of a path in the tree T, these two nodes happen to be x_j and x'_j – so that $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ – with non-negligible probability $1/(2N-1)^2 > 1/4N^2$.

Therefore, with probability at least $1/4N^2$, the value $D_2^\star$ is precisely the element $D_2^\dagger$ of the challenge private key $d_{\mathsf{ID}^\dagger}$ sent by the key-robustness challenger. Also, we note that $d_{\mathsf{ID}^\diamond}$ and $(D_1^\star, D_2^\star)$ necessarily correspond to distinct identities as the signature $\sigma^\star$ would not trace to a revoked user otherwise. If the desirable event $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ comes about, this implies that either:

- $(D_1^\star, D_2^\star) = (D_1^\dagger, D_2^\dagger)$, which means that $d_{\mathsf{ID}^\dagger}$ and $(D_1^\star, D_2^\star)$ correspond to distinct hierarchical identities $\mathsf{ID}^\dagger = (I_1, \ldots, I_{\ell_2-\ell_1+2})$ and $(I'_1, \ldots, I_k)$, with $k \in \{1, \ldots, \ell\}$, such that

$$h_0 \cdot \prod_{i=1}^{\ell_2-\ell_1+2} h_i^{I_i} = h_0 \cdot \prod_{i=1}^{k'} h_i^{I'_i}.$$

  Such a collision is known (as shown in [43][Section 1.2], for example) to occur with negligible probability under the discrete logarithm assumption.
- $D_2^\star = D_2^\dagger$ and $D_1^\star \neq D_1^\dagger$, in which case $\mathcal{B}$ wins the selective key-robustness game. It does so by outputting the decryption component $(g_2^\alpha \cdot D_1^\star, D_2^\star)$ – after having extracted $(D_1^\star, D_2^\star)$ from $\{com_{D_i}^\star\}_{i=1}^2$ using (β_1, β_2) – and the identity ID' corresponding to the HIBE ciphertext $C_l^\star$ of the revocation list. Indeed, if $\mathcal{B}$ correctly guessed x_j and x'_j, ID' cannot be a descendant of $\mathsf{ID}^\dagger$ as long as $\sigma^\star$ opens to a revoked user in $U^a \cap \mathcal{R}_{t^\star}$.

Since the probability to have $\mathsf{ID}^\diamond = \mathsf{ID}^\dagger$ is at least $1/4N^2$ and due to the multiplicative factor $(1 - 1/p)$ in the statement of Lemma 1, the announced result follows. □

D Security against Framing Attacks and Anonymity

D.1 Framing Attacks

Theorem 2 (Non-frameability). *The scheme is secure against framing attacks assuming that: (i) the q_b-SDH assumption holds in $\mathbb{G}$, where q_b is the maximal number of $Q_{\mathsf{b\text{-}join}}$-queries; (ii) Σ is a strongly unforgeable one-time signature.*

Proof. As in [39], we consider two kinds of framing attacks that can be possibly mounted by a non-frameability adversary $\mathcal{A}$.

- **Type I attacks**: the adversary $\mathcal{A}$ generates a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was used by some honest group member $i \in U^b$ when answering a Q_{sig}-query.
- **Type II attacks**: $\mathcal{A}$ outputs a forgery $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$ for which the one-time verification key $\mathsf{VK}^\star$ was never used by Q_{sig} to answer a signing query on behalf of an honest user $i \in U^b$.

It is immediate that Type I attacks imply a breach in the unforgeability of the one-time signature. Lemma 3 shows that no PPT adversary can produce a Type II forgery as long as the Strong Diffie-Hellman assumption holds. □

Lemma 3. *The scheme is secure against framing attacks of Type II if the q_s-SDH problem is hard. More precisely, the advantage of any adversary after q_s Q_{sig}-queries and q_b $Q_{\mathsf{b\text{-}join}}$-queries is at most* $\mathbf{Adv}^{\text{fra-II}}(\lambda) \leq q_b \cdot \mathbf{Adv}^{\text{q}_\text{s}\text{-SDH}}(\lambda)$.

Proof. By hypothesis, the adversary $\mathcal{A}$ comes up with a forgery $(M^\star, \sigma^\star)$ that opens to some honest user $i \in U^b$ and that did not issue a signature containing the verification key $\mathsf{VK}^\star$. The same proof as in [39] shows that the Strong Diffie-Hellman assumption can be broken.

Given a problem instance $(\tilde{g}, \tilde{g}^a, \ldots, \tilde{g}^{(a^{q_s})}) \in \mathbb{G}^{q_s+1}$, the simulator $\mathcal{B}$ generates q_s one-time signature keys pairs $(\mathsf{SK}_i, \mathsf{VK}_i) \leftarrow \mathcal{G}(\lambda)$ for $i = 1$ to q_s. Then, using standard techniques (see [13][Lemma 3.2]) it builds a generator g and a randomly distributed public value $X^\dagger = g^a$ – which implicitly defines $x^\dagger = \log_g(X^\dagger) = a$ – such that it knows $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$.

Next, using the newly generated g, $\mathcal{B}$ generates key pairs $\{(sk_{\mathsf{AHO}}^{(b)}, pk_{\mathsf{AHO}}^{(b)})\}_{b=0,1}$ for the AHO signature (note that group elements of $\{pk_{\mathsf{AHO}}^{(b)}\}_{b=0,1}$ are computed as powers of g) and uses $pk_{\mathsf{AHO}}^{(0)}, pk_{\mathsf{AHO}}^{(1)}$ to form the group public key

$$\mathcal{Y} = \big(g,\ pk_{\mathsf{AHO}}^{(0)},\ pk_{\mathsf{AHO}}^{(1)},\ mpk_{\mathsf{BBG}},\ \mathbf{f},\ U,\ V,\ \mathcal{H},\ \Sigma\big).$$

In the latter, the Groth-Sahai CRS $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ is prepared for the perfect soundness setting, *i.e.*, with $\vec{f}_1 = (f_1 = g^{\beta_1}, 1, g)$, $\vec{f}_2 = (1, f_2 = g^{\beta_2}, g)$ and $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$, where $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$.

Should the adversary $\mathcal{A}$ decide to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ has $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$ and $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2) = (\log_g(f_1), \log_g(f_2))$ at its disposal. At the beginning of the game, $\mathcal{B}$ picks a random index $j^\star \xleftarrow{R} \{1, \ldots, q_b\}$ and interacts with $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: if $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ surrenders $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$.
- $Q_{\mathsf{b\text{-}join}}$-queries: when $\mathcal{A}$, acting as a rogue group manager, requests the introduction of a new honest user i in the group, $\mathcal{B}$ starts interacting with $\mathcal{A}$ in an execution of Join and runs $\mathsf{J}_{\mathsf{user}}$ on behalf of the prospective user. Namely, $\mathcal{B}$'s behavior depends on the index $j \in \{1, \ldots, q_b\}$ of the $Q_{\mathsf{b\text{-}join}}$-query.
  - If $j \neq j^\star$, $\mathcal{B}$ follows exactly the specification of $\mathsf{J}_{\mathsf{user}}$.
  - If $j = j^\star$, $\mathcal{B}$ sends the value $X^\dagger$ to J_{GM} at step 1 of Join. This implicitly defines user $j^\star$'s membership secret to be the unknown exponent $\mathsf{sec}_{j^\star} = a$ of the SDH instance. In subsequent steps of the join protocol, $\mathcal{B}$ proceeds as the real $\mathsf{J}_{\mathsf{user}}$ would. When Join terminates, $\mathcal{B}$ obtains a membership certificate $\mathsf{cert}_{j^\star} = \big(\langle v_j \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X^\dagger\big)$.
- Q_{pub}-queries: can be treated as in the real game, by having the simulator return $\mathcal{Y}$.
- Q_{sig}-queries: when $\mathcal{A}$ asks user $i \in U^b$ to sign a message M, the simulator $\mathcal{B}$ can answer the query by running the real signature generation algorithm if $i \neq j^\star$. Otherwise (namely, if $i = j^\star$), $\mathcal{B}$ uses the next available pair $\{(g^{1/(a+\mathsf{VK}_i)}, \mathsf{VK}_i)\}_{i=1}^{q_s}$ to define σ_{VK_i}. It also recalls the membership certificate $\mathsf{cert}_{j^\star} = \big(\langle v_j \rangle, \{\{d_w, \sigma_w\}_{w \in \mathsf{copath}_{x_j}}\}_{j=0}^{\ell}, X^\dagger\big)$ that it obtained from the J_{GM}-executing adversary at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query. It is easy to see that, using σ_{VK_i} and $\mathsf{cert}_{j^\star}$, it can easily generate all signature components and sign them all using SK_i.

Finally, $\mathcal{A}$ outputs a signature $\sigma^\star = \big(\mathsf{VK}^\star, \Psi_1^\star, \Psi_2^\star, \Psi_3^\star, \Psi_4^\star, \Psi_5^\star, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma_{ots}^\star\big)$, for some message $M^\star$, that opens to some user $i^\star \in U^b$ who did not sign $M^\star$. At this point, $\mathcal{B}$ halts and declares failure if it turns out that $X^\dagger \neq \Psi_3^\star \cdot \Psi_1^{\star -1/\beta_1} \cdot \Psi_2^{\star -1/\beta_2}$ since, in this case, it was unfortunate when drawing the random index $j^\star$. Still, with probability $1/q_b$, the signature $\sigma^\star$ opens to the user introduced at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query and $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ does decrypt to $X^\star$. In this situation, the perfect soundness of the proof system ensures that $com_{\sigma_{\mathsf{VK}^\star}}^\star$ is a commitment to a group element $\sigma_{\mathsf{VK}^\star}^\star$ such that $e(\sigma_{\mathsf{VK}^\star}^\star, X^\dagger \cdot g^{\mathsf{VK}^\star}) = e(g, g)$. Since $\sigma^\star$ is a Type II forgery, $\mathcal{B}$ can use β_1, β_2 to compute a BBS decryption of $com_{\sigma_{\mathsf{VK}^\star}}^\star$ and obtain a solution $(\sigma_{\mathsf{VK}^\star}, \mathsf{VK}^\star)$ to the q_b-SDH instance. □

D.2 Anonymity

As for the anonymity property, it naturally relies on the DLIN assumption. The proof is essentially identical to that of Lemma 5 in [39] but we give it for completeness.

Theorem 3 (Anonymity). *The advantage of any anonymity adversary is at most*

$$\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda),$$

where the first term is $\mathcal{A}$'s probability of breaking the strong unforgeability of the one-time signature.

Proof. We consider a sequence of games at the end of which even an unbounded adversary has no advantage. In Game i, we call S_i the event that $\mathcal{A}$ wins and define $Adv_i = |\Pr[S_i] - 1/2|$.

Game 1: is the experiment of definition 6. In the play stage, the adversary $\mathcal{A}$ can obtain the group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}} = (sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}})$. It can also ask for the opening of any group signature and read/write the content of $\mathsf{state}_{\mathcal{I}}$. When it decides to enter the challenge phase, it outputs a message $M^\star$, a period index $t^\star$ and two membership certificate/secret $(\mathsf{cert}^\star_0, \mathsf{sec}^\star_0)$ and $(\mathsf{cert}^\star_1, \mathsf{sec}^\star_1)$ such that $\mathsf{cert}^\star_b \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}^\star_b$ for $b = 0, 1$. The simulator $\mathcal{B}$ flips a fair coin $d \xleftarrow{R} \{0,1\}$ and computes $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, t^\star, RL_{t^\star}, \mathsf{cert}^\star_d, \mathsf{sec}^\star_d, M^\star)$, where $t^\star$ is determined by the history of Q_{revoke}-queries. The signature $\sigma^\star$ is given as a challenge to $\mathcal{A}$ who has to guess $d \in \{0,1\}$ after another series of queries (under the natural restriction of not querying the opening of $\sigma^\star$). We have $Adv_1 = \mathbf{Adv}^{\mathrm{anon}}(\mathcal{A})$.

Game 2: is as **Game** 1 but $\mathcal{B}$ halts if $\mathcal{A}$ queries the opening of a signature σ containing the same one-time verification key $\mathsf{VK}^\star$ as in the challenge phase (we assume w.l.o.g. that $(\mathsf{SK}^\star, \mathsf{VK}^\star)$ is generated at the outset of the game). If such a query is made before the challenge phase, it means that $\mathcal{A}$ was able to forge a one-time signature even without having seen a signature. If the query occurs after the challenge phase, then the strong unforgeability of Σ is broken. We can thus write $|\Pr[S_2] - \Pr[S_1]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$.

Game 3: we change the generation of $\mathcal{Y}$ so as to answer Q_{open}-queries without using the secret exponents $\beta_1, \beta_2 \in \mathbb{Z}_p$ that define $\mathcal{S}_{\mathsf{OA}}$. To this end, $\mathcal{B}$ chooses $\alpha_u, \alpha_v \xleftarrow{R} \mathbb{Z}_p^*$, and defines $U = g^{-\mathsf{VK}^\star} \cdot f_1^{\alpha_u}$, and $V = g^{-\mathsf{VK}^\star} \cdot f_2^{\alpha_v}$. It is not hard to see (see [47] for details) that, for any Q_{open}-query containing a BBS encryption $(\Psi_1, \Psi_2, \Psi_3) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2})$, the values (Ψ_4, Ψ_5) reveal g^{z_1} and g^{z_2} (and thus the encrypted X) since $\mathsf{VK} \neq \mathsf{VK}^\star$ unless the event introduced in Game 2 occurs. To generate the challenge signature $\sigma^\star$ at epoch $t^\star$, $\mathcal{B}$ first computes $(\Psi^\star_1, \Psi^\star_2, \Psi^\star_3)$ and then $(\Psi^\star_4, \Psi^\star_5) = (\Psi^{\star\alpha_u}_1, \Psi^{\star\alpha_v}_2)$. It sets the challenge signature to be $\sigma^\star = (\mathsf{VK}^\star, \Psi^\star_1, \Psi^\star_2, \Psi^\star_3, \Psi^\star_4, \Psi^\star_5, \Omega^\star, \mathbf{com}^\star, \mathbf{\Pi}^\star, \sigma^\star_{ots})$. It can be checked that the distributions of $\mathcal{Y}$ and $\sigma^\star$ are unchanged and we have $\Pr[S_3] = \Pr[S_2]$.

Game 4: in the setup phase, we generate the CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ of the proof system for the perfect WI setting. We choose $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2} \odot (1,1,g)^{-1}$ instead of $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$ so that $\vec{f_1}$, $\vec{f_2}$ and $\vec{f_3}$ are linearly independent. Any significant change in $\mathcal{A}$'s behavior yields a distinguisher for the DLIN problem and we can write $|\Pr[S_4] - \Pr[S_3]| = 2 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. As noted in [40], proofs in the WI setting reveal no information on which witnesses they were generated from.

Game 5: we modify the generation of the challenge $\sigma^\star$ and use the trapdoor of the CRS (*i.e.*, ξ_1, ξ_2 s.t. $\vec{\varphi} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$) to simulate proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ that $(\Psi^\star_1, \Psi^\star_2, \Psi^\star_3)$ and com_X encrypt of the same value. It is known [40] that linear multi-exponentiation equations always have perfectly NIZK proofs on a simulated CRS. For, any satisfiable relation, (ξ_1, ξ_2) allows generating proofs without using the witnesses τ_1, τ_2, τ_3 for which (9) holds and simulated proofs are perfectly indistinguishable from real ones. Hence, $\Pr[S_5] = \Pr[S_4]$.

Game 6: in the computation of $\Psi^\star_3$, we now replace $g^{z_1+z_2}$ by a random group element in the challenge $\sigma^\star$. Since $\mathcal{B}$ does not explicitly use $z_1 = \log_{f_1}(\Psi^\star_1)$, $z_2 = \log_{f_2}(\Psi^\star_2)$, any change in $\mathcal{A}$'s behavior

yields a distinguisher for the DLIN problem and $|\Pr[S_6] - \Pr[S_5]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. In Game 6, we have $\Pr[S_6] = 1/2$. Indeed, when we consider the challenge $\sigma^\star$, Groth-Sahai commitments are all perfectly hiding in the WI setting and proofs $\mathbf{\Pi}$ reveal nothing about the underlying witnesses (in particular, NIZK proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^3$ are generated without using them) and $(\Psi_1^\star, \Psi_2^\star, \Psi_3^\star)$ perfectly hides $X^\star$. Finally, randomized signature components $\Omega^\star = \{\Theta'^\star_{l,i}, \theta'^\star_{l,i}\}_{i\in\{3,4,6,7\}}$ are information-theoretically independent of the corresponding messages and the remaining components of AHO signatures $\Theta_l^\star$ and $\theta_l^\star$.

When combining the above, $\mathcal{A}$'s advantage can be bounded by $\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3\cdot\mathbf{Adv}^{\mathrm{DLIN}}(\lambda)$ as stated by the theorem. □

E A Construction Based on the Complete Subtree Method

The following construction uses the public-key variant (suggested in [51, 34]) of the CS method, which does not require a hierarchical IBE: a single-level selectively secure IBE scheme such as the one described by Boneh and Boyen [14] suffices. As in our construction based on the SD method, we do not need to use the master secret key of the IBE system.

In the upcoming description, the main difference with the scheme of section 3 is the way to distribute IBE private keys in the join protocol. Other algorithms are essentially unchanged.

As in section 3, the number of users is assumed to be $N = 2^\ell$ so that each group member is assigned to a leaf of the tree. Again, each node is assigned a unique identifier. For simplicity, we define the identifier $\mathsf{ID}(x) \in \{1, \ldots, 2N-1\}$ of node x to be $\mathsf{ID}(x) = 2 \cdot \mathsf{ID}(\mathsf{parent}(x)) + b$, where $\mathsf{parent}(x)$ denotes x's father in the tree and $b = 0$ (resp. $b = 1$) if x is the left (resp. right) child of its father. The root of the tree is assigned the identifier $\mathsf{ID}_\epsilon = 1$.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N = 2^\ell$,

1. Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, with a generator $g \xleftarrow{R} \mathbb{G}$.
2. Generate two key pairs $(sk^{(0)}_{\mathsf{AHO}}, pk^{(0)}_{\mathsf{AHO}})$ and $(sk^{(1)}_{\mathsf{AHO}}, pk^{(1)}_{\mathsf{AHO}})$ for the AHO signature in order to sign messages of two group elements. These key pairs consist of

$$pk^{(d)}_{\mathsf{AHO}} = \Big(G_r^{(d)},\ H_r^{(d)},\ G_z^{(d)} = G_r^{\gamma_z^{(d)}},\ H_z^{(d)} = H_r^{\delta_z^{(d)}},\ \{G_i^{(d)} = G_r^{\gamma_i^{(d)}}, H_i^{(d)} = H_r^{\delta_i^{(d)}}\}_{i=1}^2,\ A^{(d)},\ B^{(d)}\Big)$$

   and $sk^{(d)}_{\mathsf{AHO}} = \big(\alpha_a^{(d)}, \alpha_b^{(d)}, \gamma_z^{(d)}, \delta_z^{(d)}, \{\gamma_i^{(d)}, \delta_i^{(d)}\}_{i=1}^2\big)$, where $d \in \{0, 1\}$.
3. As a CRS for the NIWI proof system, select vectors $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ s.t. $\vec{f_1} = (f_1, 1, g) \in \mathbb{G}^3$, $\vec{f_2} = (1, f_2, g) \in \mathbb{G}^3$, and $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$, with $f_1 = g^{\beta_1}, f_2 = g^{\beta_2} \xleftarrow{R} \mathbb{G}$ and $\beta_1, \beta_2, \xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$.
4. Choose $(U, V) \xleftarrow{R} \mathbb{G}^2$ that, together with f_1, f_2, g, will form a public key for an IND-CCA2 cryptosystem.
5. Generate a master public key mpk_{BB} for the Boneh-Boyen IBE. Such a public key consists of $mpk_{\mathsf{BB}} = \big(h_0, h_1\big)$ and, again, no master secret key is needed.
6. Select a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$.
7. Set $\mathcal{S}_{\mathsf{GM}} := \big(sk^{(0)}_{\mathsf{AHO}}, sk^{(1)}_{\mathsf{AHO}}\big)$, $\mathcal{S}_{\mathsf{OA}} := \big(\beta_1, \beta_2\big)$ as authorities' private keys and define the group public key to be

$$\mathcal{Y} := \Big(g,\ pk^{(0)}_{\mathsf{AHO}},\ pk^{(1)}_{\mathsf{AHO}},\ mpk_{\mathsf{BB}},\ \mathbf{f},\ (U, V),\ \Sigma\Big).$$

Join$^{(\mathrm{GM},\mathcal{U}_i)}$**:** the group manager and the prospective user $\mathcal{U}_i$ carry out the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ chooses $x \xleftarrow{R} \mathbb{Z}_p$ and computes $X = g^x$ which is sent to $\mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})$. If the value X already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} aborts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} assigns to $\mathcal{U}_i$ an available leaf v_i of the tree T and we let $\mathsf{ID}(v_i)$ be the identifier of v_i. Let $x_0 = \epsilon$, x_1, ..., $x_{\ell-1}$, $x_\ell = v_i$ be the path connecting the leaf v_i to the root ϵ of T. For $j = 0$ to ℓ, J_{GM} conducts the following steps.
   a. Compute an IBE private key $D_{x_j} = (D_{x_j,1}, D_{x_j,2}) = \Big(\big(h_0^{\mathsf{ID}(x_j)} \cdot h_1\big)^{r_{x_j}},\ g^{r_{x_j}}\Big)$ using a randomly chosen $r_{x_j} \xleftarrow{R} \mathbb{Z}_p$.
   b. Generate an AHO signature $\sigma_{x_j} = (\theta_{x_j,1}, \ldots, \theta_{x_j,7})$ on the pair $\big(X, D_{x_j,2}\big) \in \mathbb{G}^2$ so as to bind the node x_j and the value X that identifies $\mathcal{U}_i$.
3. J_{GM} sends the IBE private keys $\{D_{x_j}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ that verifies their validity. If all keys are well-formed, $\mathsf{J}_{\mathsf{user}}$ acknowledges these values by generating a digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}\big(X||\{D_{x_j}\}_{j=0}^{\ell}\big)$ and sends it back to J_{GM}.
4. J_{GM} checks that $\mathrm{Verify}_{\mathsf{upk}[i]}\big(X||\{D_{x_j}\}_{j=0}^{\ell}, sig_i\big) = 1$. If not J_{GM} aborts. Otherwise, J_{GM} sends the AHO signatures $\{\sigma_{x_j}\}_{j=0}^{\ell}$ to $\mathsf{J}_{\mathsf{user}}$ and stores $\mathsf{transcript}_i = (X, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, sig_i)$ in St_{trans}.
5. $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate cert_i as $\mathsf{cert}_i = \big(\langle v_i \rangle, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, X\big)$. The membership secret sec_i is defined to be $\mathsf{sec}_i = x$.

Revoke$(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, t, \mathcal{R}_t)$**:**

1. Parse $\mathcal{S}_{\mathsf{GM}}$ as $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{AHO}}^{(0)}, sk_{\mathsf{AHO}}^{(1)})$.
2. Using the CS covering algorithm, find a cover of the unrevoked user set $\{1, \ldots, N\}\backslash\mathcal{R}_t$ as the union of m sub-trees $S_1, \ldots, S_m$, with $m \leq r \cdot \log(N/r)$. Let $u_1, \ldots, u_m$ be the roots of these sub-trees
3. For $i = 1$ to m, do the following.
   a. Compute an IBE ciphertext $C_i = \big(h_0^{\mathsf{ID}(u_i)} \cdot h_1\big)$ for the identity $\mathsf{ID}(u_i)$.
   b. To authenticate C_i and bind it to the current revocation epoch t, use $sk_{\mathsf{AHO}}^{(1)}$ to generate an AHO signature $\Theta_i = (\Theta_{i,1}, \ldots, \Theta_{i,7}) \in \mathbb{G}^7$ on the pair $(C_i, g^t) \in \mathbb{G}^2$, where the epoch number t is interpreted as an element of $\mathbb{Z}_p$.

Return the revocation data RL_t which is defined to be

$$RL_t = \Big(t,\ \mathcal{R}_t,\ \{\mathsf{ID}(u_i), (C_i, \Theta_i)\}_{i=1}^{m}\Big) \tag{14}$$

Sign$(\mathcal{Y}, t, RK_t, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** return $\perp$ if $i \in \mathcal{R}_t$. Otherwise, to sign $M \in \{0,1\}^*$, generate a one-time key pair $(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(\lambda)$. Parse cert_i and sec_i as $\big(\langle v_i \rangle, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, X\big)$ and $x \in \mathbb{Z}_p$, respectively. Then, $\mathcal{U}_i$ conducts the following steps.

1. Using RL_t, determine the sub-tree S_l with $l \in \{1, \ldots, m\}$, that contains the leaf v_i (this subset must exist since $i \notin \mathcal{R}_t$) and let u_l be the root of S_l. Since u_l is an ancestor of v_i, the signer $\mathcal{U}_i$ necessarily knows and IBE private key of the form

$$D_{u_l} = (D_{u_l,1}, D_{u_l,2}) = \Big(\big(h_0^{\mathsf{ID}(u_l)} \cdot h_1\big)^{r_{u_l}},\ g^{r_{u_l}}\Big). \tag{15}$$

2. To prove that he holds a valid IBE private key for $C_l = \big(h_0^{\mathsf{ID}(u_l)} \cdot h_1\big)$, $\mathcal{U}_i$ first generates a commitment com_{C_l} to C_l. Then, he re-randomizes the corresponding signature $\Theta_l = (\Theta_{l,1}, \ldots, \Theta_{l,7})$ to

obtain $\{\Theta'_{l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk^{(1)}_{\mathsf{AHO}}, \Theta_l)$ and computes commitments $\{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}$ to the resulting $\{\Theta'_{l,i}\}_{i\in\{1,2,5\}}$. Finally, he generates a proof π_{C_l} that C_l is a certified HIBE ciphertext for epoch t: *i.e.*, π_{C_l} provides evidence that

$$\begin{aligned} A^{(1)} \cdot e(\Theta'_{l,3}, \Theta'_{l,4})^{-1} \cdot e(G_2^{(1)}, g^t)^{-1} &= e(G_z^{(1)}, \Theta'_{l,1}) \cdot e(G_r^{(1)}, \Theta'_{l,2}) \cdot e(G_1^{(1)}, C_l), \\ B^{(1)} \cdot e(\Theta'_{l,6}, \Theta'_{l,7})^{-1} \cdot e(H_2^{(1)}, g^t)^{-1} &= e(H_z^{(1)}, \Theta'_{l,1}) \cdot e(H_r^{(1)}, \Theta'_{l,5}) \cdot e(H_1^{(1)}, C_l), \end{aligned} \tag{16}$$

Then, $\mathcal{U}_i$ generates commitments $com_{D_{u_l,1}}$ and $com_{D_{u_l,2}}$ to $D_{u_l,1}$ and $D_{u_l,2}$. Then, he generates a proof $\pi_{D_{u_l}}$ that $e(D_{u_l,1}, g) = e(C_l, D_{u_l,2})$. Since $\{\Theta'_{l,i}\}_{i\in\{3,4,6,7\}}$ are constants, the two relations of (16) are linear equations and π_{C_l} costs 6 elements while $\pi_{D_{u_l}}$ takes 9 elements.

3. Let $\sigma_{u_l} = (\theta_{u_l,1}, \ldots, \theta_{u_l,7}) \in \mathbb{G}^7$ be the structure-preserving signature on $(X, D_{u_l,2})$. Re-randomize σ_{u_l} to obtain $\{\theta'_{u_l,i}\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk^{(0)}_{\mathsf{AHO}}, \sigma_{u_l})$. Then, generate commitments $\{com_{\theta'_{u_l,i}}\}_{i\in\{1,2,5\}}$ to $\{\theta'_{u_l,i}\}_{i\in\{1,2,5\}}$ as well as a commitment com_X to X. Finally, generate a proof $\pi_{\sigma_{u_l}}$ that committed variables $\{\theta'_{u_l,i}\}_{i\in\{1,2,5\}}$, X and $D_{u_l,2}$ satisfy the verification equations

$$\begin{aligned} A^{(0)} \cdot e(\theta'_{u_l,3}, \theta'_{u_l,4})^{-1} &= e(G_z^{(0)}, \theta'_{u_l,1}) \cdot e(G_r^{(0)}, \theta'_{u_l,2}) \cdot e(G_1^{(0)}, X) \cdot e(G_2^{(0)}, D_{u_l,2}), \\ B^{(0)} \cdot e(\theta'_{u_l,6}, \theta'_{u_l,7})^{-1} &= e(H_z^{(0)}, \theta'_{u_l,1}) \cdot e(H_r^{(0)}, \theta'_{u_l,5}) \cdot e(H_1^{(0)}, X) \cdot e(H_2^{(0)}, D_{u_l,2}). \end{aligned}$$

Since these equations are linear, $\pi_{\sigma_{u_l}}$ requires 6 group elements.

4. Compute a tag-based encryption of X by drawing $z_1, z_2 \xleftarrow{R} \mathbb{Z}_p$ and setting

$$(\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5) = \big(f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2}, (g^{\mathsf{VK}} \cdot U)^{z_1}, (g^{\mathsf{VK}} \cdot V)^{z_2}\big).$$

5. Generate a NIZK proof that the commitment $com_X = (1, 1, X) \cdot \vec{f}_1^{\,\phi_{X,1}} \cdot \vec{f}_2^{\,\phi_{X,2}} \cdot \vec{f}_3^{\,\phi_{X,3}}$ and (Ψ_1, Ψ_2, Ψ_3) are BBS encryptions of the same value X. If we write $\vec{f}_3 = (f_{3,1}, f_{3,2}, f_{3,3})$, com_X can be written as $(f_1^{\phi_{X,1}} \cdot f_{3,1}^{\phi_{X,3}}, f_2^{\phi_{X,2}} \cdot f_{3,2}^{\phi_{X,3}}, X \cdot g^{\phi_{X,1}+\phi_{X,2}} \cdot f_{3,3}^{\phi_{X,3}})$ and, given that $(\Psi_1, \Psi_2, \Psi_3) = (f_1^{z_1}, f_2^{z_2}, X \cdot g^{z_1+z_2})$, we have

$$com_X \odot (\Psi_1, \Psi_2, \Psi_3)^{-1} = \big(f_1^{\tau_1} \cdot f_{3,1}^{\tau_3},\ f_2^{\tau_2} \cdot f_{3,2}^{\tau_3},\ g^{\tau_1+\tau_2} \cdot f_{3,3}^{\tau_3}\big) \tag{17}$$

with $\tau_1 = \phi_{X,1} - z_1$, $\tau_2 = \phi_{X,2} - z_2$, $\tau_3 = \phi_{X,3}$. The signer $\mathcal{U}_i$ commits to the exponents $\{\tau_i\}_{i=1}^{3}$ (by computing $com_{\tau_j} = \vec{\varphi}^{\tau_j} \cdot \vec{f}_1^{\,\phi_{\tau_j,1}} \cdot \vec{f}_2^{\,\phi_{\tau_j,2}}$ for $j \in \{1, 2, 3\}$, using the vector $\vec{\varphi} = \vec{f}_3 \cdot (1, 1, g)$), and generates proofs $\pi_{eq\text{-}com,1}$, $\pi_{eq\text{-}com,2}$ and $\pi_{eq\text{-}com,3}$ that $\{\tau_i\}_{i=1}^{3}$ satisfy the relations (17). Since (17) are linear equations, proofs $\{\pi_{eq\text{-}com,j}\}_{j=1}^{3}$ cost 2 elements each.

6. Compute $\sigma_{\mathsf{VK}} = g^{1/(x+\mathsf{VK})}$ and generate a commitment $com_{\sigma_{\mathsf{VK}}}$ to σ_{VK}. Then, generate a NIWI proof $\pi_{\sigma_{\mathsf{VK}}}$ that committed variables σ_{VK} and X satisfy

$$e(\sigma_{\mathsf{VK}}, X \cdot g^{\mathsf{VK}}) = e(g, g) \tag{18}$$

Relation (18) is a quadratic pairing product equation and requires a proof consisting of 9 group elements.

7. Using SK, generate a one-time signature $\sigma_{ots} = \mathcal{S}(\mathsf{SK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}))$ where $\Omega = \{\Theta'_{l,i}, \theta'_{u_l,i}\}_{i\in\{3,4,6,7\}}$ and

$$\begin{aligned} \mathbf{com} &= \big(com_{C_l}, \{com_{D_{u_l,j}}\}_{j=1}^{2}, com_X, \{com_{\Theta'_{l,i}}\}_{i\in\{1,2,5\}}, \{com_{\theta'_{u_l,j}}\}_{j\in\{1,2,5\}}, \{com_{\tau_j}\}_{j=1}^{3}, com_{\sigma_{\mathsf{VK}}}\big) \\ \mathbf{\Pi} &= (\pi_{C_l}, \pi_{D_{u_l}}, \pi_{\sigma_{u_l}}, \pi_{eq\text{-}com,1}, \pi_{eq\text{-}com,2}, \pi_{eq\text{-}com,3}, \pi_{\sigma_{\mathsf{VK}}}). \end{aligned}$$

Return the signature

$$\sigma = \big(\mathsf{VK}, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}, \sigma_{ots}\big). \quad (19)$$

Verify$(\sigma, M, t, RL_t, \mathcal{Y})$**:** parse σ as above and return 1 if and only if the following checks all succeed.

1. If $\mathcal{V}(\mathsf{VK}, (M, RL_t, \Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5, \Omega, \mathbf{com}, \mathbf{\Pi}), \sigma_{ots}) = 0$, return 0.
2. Return 0 if $e(\Psi_1, g^{\mathsf{VK}} \cdot U) \neq e(f_1, \Psi_4)$ or $e(\Psi_2, g^{\mathsf{VK}} \cdot V) \neq e(f_2, \Psi_5)$.
3. Return 1 if all proofs properly verify. Otherwise, return 0.

Open$(M, t, RL_t, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** given $\mathcal{S}_{\mathsf{OA}} = (\beta_1, \beta_2)$, parse the signature σ as in (19) and return $\perp$ if $\mathsf{Verify}(\sigma, M, t, RL_t, \mathcal{Y}) = 0$. Otherwise, compute $\tilde{X} = \Psi_3 \cdot \Psi_1^{-1/\beta_1} \cdot \Psi_2^{-1/\beta_2}$. Find a record

$$\langle i, \mathsf{transcript}_i = (X, \{D_{x_j}, \sigma_{x_j}\}_{j=0}^{\ell}, sig_i)\rangle$$

such that $X = \tilde{X}$. If no such record exists in St_{trans}, return $\perp$. Otherwise, return i.

The size of signatures is exactly the same as in the construction based on the SD method. Revocation lists have become longer: they now contain $O(r \cdot \log(N/r))$ group elements (as in Section 3, the representation of $\mathsf{ID}(u_i)$ is at least as short as that of a group element in RL_t) and they can be seen as ciphertexts in the public-key variant [34] of the CS method. On the other hand, we note that membership certificates now consist of $O(\log N)$ group elements (vs $O(\log^3 N)$ in the SD method).

The complexity of the verification algorithm does not depend on r or N. As for the signing algorithm, it first requires $O(\log \log N)$ combinatorial operations (see [51]) to determine which sub-tree the signer is a leaf of. However, the cost of these operations (which are only needed once per epoch) is small compared to that of public-key arithmetic operations. As we can see, the number of arithmetic operations is independent of r and N when it comes to generate or verify signatures.

E.1 Security

All security proofs go through essentially without changes. The proof of Theorem 4 relies on the key-robustness of the Boneh-Boyen IBE [14], but this property is implied by Lemma 1: indeed, the first IBE scheme of [13] (in its single-level variant) can be seen as a single-level variant of the Boneh-Boyen-Goh HIBE.

Theorem 4 (Misidentification). *The scheme is secure against misidentification attacks assuming that the q-SFP problem is hard for $q = \max(\ell \cdot q_a, q_r^2)$, where q_a and q_r denote the maximal numbers of $Q_{\mathsf{a\text{-}join}}$ queries and Q_{revoke} queries, respectively, and $\ell = \log N$.*

Proof. The proof is almost identical to that of Theorem 1. The only difference is that, in the treatment of Type II.a forgeries, the simulator $\mathcal{B}$ has to generate at most $\ell \cdot q_a$ AHO signatures overall (rather than $\ell^2 \cdot q_a$ in the proof of Theorem 1). □

Theorem 5 (Non-frameability). *The scheme is secure against framing attacks assuming that: (i) the q_b-SDH assumption holds in $\mathbb{G}$, where q_b is the maximal number of $Q_{\mathsf{b\text{-}join}}$-queries; (ii) Σ is a strongly unforgeable one-time signature.*

Proof. The proof is the same as the proof of Theorem 2. □

Theorem 6 (Anonymity). *The advantage of any anonymity adversary is at most*

$$\mathbf{Adv}^{\mathrm{anon}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda) + 3 \cdot \mathbf{Adv}^{\mathrm{DLIN}}(\lambda).$$

Proof. The proof is completely identical to the proof of Theorem 3. □