Full-Domain Subgroup Hiding and Constant-Size Group Signatures

Xavier Boyen [1] and Brent Waters* [2]

[1] Voltage Inc., Palo Alto — xb@boyen.org
[2] SRI International — bwaters@csl.sri.com

Abstract. We give a short constant-size group signature scheme, which we prove fully secure under reasonable assumptions in bilinear groups, in the standard model. We achieve this result by using a new NIZK proof technique, related to the BGN cryptosystem and the GOS proof system, but that allows us to hide integers from the full domain rather than individual bits.

1 Introduction

Group signatures, introduced by Chaum and van Heyst [18], allow any member of a certain group to sign a message on behalf of the group, but the signer remains anonymous within the group. However, in certain extenuating circumstances an authority will have the ability to revoke the anonymity of a signer and trace the signature. One of the primary motivating use scenarios of group signatures is in anonymous attestation, which has practical applications such as in building Trusted Platform Modules (TPMs). Group signatures have also attracted much attention in the research community where several constructions have been proposed [2, 3, 15, 14, 13, 28, 8, 24, 26, 6, 5, 12, 1].

The most efficient group signature constructions given only have a proof of security in the random oracles model and either are based on the Strong-RSA assumption in Z_n [2, 3, 15] or use bilinear groups [8, 10, 16]. Solutions in the standard model can be derived from general assumptions as first shown by Bellare et. al. [5].

Recently, two efficient group signature schemes were respectively proposed both by Boyen and Waters [12] and Ateniese et. al. [1] that did not use random oracles. The two solutions took different approaches and have different features.

The Boyen-Waters construction used a two-level hierarchical signature, where the first level corresponds to the signer's identity and the second level is the message to be signed. The scheme hides the actual identity in the first level by using bilinear groups of composite order and applying a mechanism from the recent Non-Interactive Zero-Knowledge (NIZK) result of Groth, Ostrovsky, and Sahai [22]. The two drawbacks of the Boyen-Waters result are that the number

* Supported by NSF CNS-0524252 and the US Army Research Office under the CyberTA Grant No. W911NF-06-1-0316.

of group elements in the signature are logarithmic in the number of signers in the group and that the anonymity property is only secure against chosen-plaintext attacks, as opposed to chosen-ciphertext attacks. The need for a logarithmic number of group elements results from the fact that a signer must prove that the blinded first level identity was computed correctly. The authors needed to use the model for CPA attacks because the tracing authority used the knowledge of the factorization of the order to trace members.

The Ateniese et. al. scheme works in asymmetric bilinear groups. Their scheme has signatures with a constant number of group elements and has chosen-ciphertext security. However, its proofs of security rely on interactive assumptions where the adversary has access to an oracle; therefore, these assumptions are inherently non-falsifiable [27]. In addition, the scheme has the drawback that if a user's private key is compromised then it can be used to revoke the anonymity of that user's past signatures. Although, it should be pointed out that some schemes have used this property as an advantage in Verifier-Local Group signatures [10].

Groth [20] also gave a recent group signature scheme that was proven CCA-secure in the standard model under the decisional-linear assumption [8]. Signatures in his scheme technically consist of a constant number of group elements, however, as noted by the author the constant is too large for real systems and in practice his constant will be much more than $\lg(n)$ for any reasonable number of n signers. The result does though, give a feasibility result under a relatively mild assumption.

In this paper we give a new construction of a group signature scheme that addresses some of the drawbacks of the Boyen-Waters [12] solution. Following their scheme we use a two-level hierarchical signature as the basis for our signatures, where the first level specifies the identity. However, we use a new signature on the first level based off an assumption related to Strong Diffie-Hellman (SDH) [7] that we call the Hidden Strong Diffie-Hellman, which like SDH and Strong-RSA has the property that the adversary has flexibility in what he is allowed to return to the challenger. The signature has the property that if the signer gives a signature on an arbitrary group element this can be used to break our assumption. We provide efficient proofs of well-formmess that use techniques beyond those given in [22], including proofs of encrypted Diffie-Hellman tuples. One disadvantage of this approach is that it uses a stronger assumption for unforgeability than CDH, which was used in the Boyen-Waters [12] scheme. However, we emphasize that this assumption is falsifiable.

2 Preliminaries

We review a number of useful notions from the recent literature on pairing-based cryptography, which we shall need in later sections. First, we briefly review the properties that constitute a group signature scheme and define its security.

We take this opportunity to clarify once and for all that, in this paper, the word "group" by default assumes its algebraic meaning, except in contexts such as "group signature" and "group manager" where it designates a collection of users. There should be no ambiguity from context.

2.1 Group Signatures

A group signature scheme consists of a pentuple of PPT algorithms:

- A group setup algorithm, *Setup*, that takes as input a security parameter 1^λ (in unary) and the size of the group, 2^k, and outputs a public key PK for verifying signatures, a master key MK for enrolling group members, and a tracing key TK for identifying signers.
- An enrollment algorithm, *Enroll*, that takes the master key MK and an identity ID, and outputs a unique identifier s_{ID} and a private signing key K_{ID} which is to be given to the user.
- A signing algorithm, *Sign*, that takes a group member's private signing key K_{ID} and a message M, and outputs a signature σ.
- A (usually deterministic) verification algorithm, *Verify*, that takes a message M, a signature σ, and a group verification key PK, and outputs either `valid` or `invalid`.
- A (usually deterministic) tracing algorithm, *Trace*, that takes a valid signature σ and a tracing key TK, and outputs an identifier s_{ID} or the failure symbol $\perp$.

There are four types of entities one must consider:

- The group master, which sets up the group and issues private keys to the users. Often, the group master is an ephemeral entity, and the master key MK is destroyed once the group is set up. Alternatively, techniques from distributed cryptography can be used to realize the group master functionality without any real party becoming in possession of the master key.
- The group manager, which is given the ability to identify signers using the tracing key TK, but not to enroll users or create new signing keys.
- Regular member users, or signers, which are each given a distinct private signing key K_{ID}.
- Outsiders, or verifiers, who can only verify signatures using the public key PK.

We require the following correctness and security properties.

Consistency. The consistency requirements are such that, whenever, (for a group of 2^k users)

$$(\mathsf{PK}, \mathsf{MK}, \mathsf{TK}) \leftarrow \mathit{Setup}(1^\lambda, 2^k),$$

$$(s_{\mathsf{ID}}, K_{\mathsf{ID}}) \leftarrow \mathit{Enroll}(\mathsf{MK}, \mathsf{ID}), \qquad \sigma \leftarrow \mathit{Sign}(K_{\mathsf{ID}}, M),$$

we have, (except with negligible probability over the random bits used in *Verify* and *Trace*)

$$\mathit{Verify}(M, \sigma, \mathsf{PK}) = \texttt{valid}, \qquad \text{and} \qquad \mathit{Trace}(\sigma, \mathsf{TK}) = s_{\mathsf{ID}}.$$

The unique identifier s_{ID} can be used to assist in determining the user ID from the transcript of the *Enroll* algorithm; s_{ID} may but need not be disclosed to the user; it may be the same as ID.

Security. Bellare, Micciancio, and Warinschi [5] characterize the fundamental properties of group signatures in terms of two crucial security properties from which a number of other properties follow. The two important properties are:

Full Anonymity which requires that no PPT adversary be able to decide (with non-negligible probability over one half) whether a challenge signature σ on a message M emanates from user ID_1 or ID_2, where ID_1, ID_2, and M are chosen by the adversary. In the original definition of [5], the adversary is given access to a tracing oracle, which it may query before and after being given the challenge σ, much in the fashion of IND-CCA2 security for encryption.

Boneh, Boyen, and Shacham [8] relax this definition by withholding access to the tracing oracle, thus mirroring the notion of IND-CPA security for encryption. We follow [8] and speak of *CCA2-full anonymity* and *CPA-full anonymity* for the respective notions.

Full Traceability which requires that no coalition of users be able to generate, in polynomial time, a signature that passes the *Verify* algorithm but fails to trace to a member of the coalition under the *Trace* algorithm. According to this notion, the adversary is allowed to ask for the private keys of any user of its choice, adaptively, and is also given the secret key TK to be used for tracing—but of course not the enrollment master key MK.

It is noted in [5] that this property implies that of *exculpability* [4], which is the requirement that no party should be able to frame a honest group member as the signer of a signature he did not make, not even the group manager. However, the model of [5] does not consider the possibility of a (long-lived) group master, which leaves it as a potential framer. To address this problem and achieve the notion of *strong exculpability*, introduced in [2] and formalized in [25, 6], one would need an interactive enrollment protocol, call *Join*, at the end of which only the user himself knows his full private key; the same mechanism may also enable concurrent dynamic group enrollment [6, 26].

We refer the reader mainly to [5] for more precise definitions of these and related notions.

2.2 Bilinear Groups of Composite Order

We review some general notions about bilinear maps and groups, with an emphasis on groups of *composite order* which will be used in most of our constructions. We follow [9] in which composite order bilinear groups were first introduced in cryptography.

Consider two finite cyclic groups G and G_T having the same order n, in which the respective group operation is efficiently computable and denoted multiplicatively. Assume that there exists an efficiently computable function $e : G \times G \to G_T$, called a bilinear map or pairing, with the following properties:

- (Bilinearity) $\forall u, v \in G$, $\forall a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$, where the product in the exponent is defined modulo n;
- (Non-degeneracy) $\exists g \in G$ such that $e(g, g)$ has order n in G_T. In other words, $e(g, g)$ is a generator of G_T, whereas g generates G.

If such a bilinear map can be computed efficiently, the group G is called a bilinear group. We remark that the vast majority of cryptosystems based on pairings assume for simplicity that bilinear groups have prime order. In our case, it is important that the pairing be defined over a group G containing $|G| = n$ elements, where $n = pq$ has a (ostensibly hidden) factorization in two large primes, $p \neq q$.

2.3 Complexity Assumptions

We shall make use of a few complexity assumptions: computational Diffie-Hellman (CDH) in the prime-order bilinear subgroup G_p, Subgroup Decision in the group G of composite order $n = pq$, and a new assumption in G_p related to Strong Diffie-Hellman (SDH) that we call HSDH.

CDH in Bilinear Groups. The CDH assumption states that there is no probabilistic polynomial time (PPT) algorithm that, given a triple $(g, g^a, g^b) \in G_p^3$ for random exponents $a, b \in \mathbb{Z}_p$, computes $g^{ab} \in G_p$ with non-negligible probability. Because of the pairing, CDH in G_p implies a "Gap DH" assumption [23] and should not be confused with the vanilla CDH assumption in usual non-pairing groups. It is also subsumed by the HSDH assumption we describe later.

The Subgroup Decision Assumption. Our second tool is the Subgroup Decision assumption introduced in [9]. It combines features of bilinear pairings with the hardness of factoring, which is the reason for working with bilinear groups of composite order.

Informally, the Subgroup Decision assumption posits that for a bilinear group G of composite order $n = pq$, the uniform distribution on G is computationally indistinguishable from the uniform distribution on a subgroup of G (say, G_q, the subgroup of order q). The precise definition is based on the subgroup decision problem, which we now define.

Consider an "instance generator" algorithm $\mathcal{GG}$ that, on input a security parameter 1^λ, outputs a tuple (p, q, G, G_T, e), in which p and q are independent uniform random λ-bit primes, G and G_T are cyclic groups of order $n = pq$ with efficiently computable group operations (over their respective elements, which must have a polynomial size representation in λ), and $e : G \times G \to G_T$ is a bilinear map. Let $G_q \subset G$ denote the subgroup of G of order q. The subgroup decision problem is:

> On input a tuple $(n = pq, G, G_T, e)$ derived from a random execution of $\mathcal{GG}(1^\lambda)$, and an element w selected at random either from G or from G_q, decide whether $w \in G_q$.

The advantage of an algorithm $\mathcal{A}$ solving the subgroup decision problem is defined as $\mathcal{A}$'s excess probability, beyond $\frac{1}{2}$, of outputting the correct solution. The probability is defined over the random choice of instance and the random bits used by $\mathcal{A}$.

The HSDH Assumption. Last, we need to introduce a new assumption we call Hidden SDH by analogy to the SDH assumption [7] from which it descends. We present it in the next section.

3 The Hidden Strong Diffie-Hellman Assumption

We introduce a new assumption in the prime-order bilinear group G_p. It is a variant of the Strong Diffie-Hellman (SDH) assumption proposed in [7]. It is slightly stronger, but retains the attributes of the original assumption of being non-interactive, falsifiable, and provably true in the generic bilinear group model.

The Strong Diffie-Hellman assumption in bilinear groups states that there is no probabilistic polynomial time (PPT) adversary that, given a $(\ell + 1)$-tuple $(g, g^{\omega}, g^{\omega^2}, \ldots, g^{\omega^{\ell}}) \in G_p^{\ell+1}$ for a random exponent $\omega \in \mathbb{Z}_p^*$, outputs a pair $(c, g^{1/(\omega+c)}) \in \mathbb{Z}_p^* \times G_p$ with non-negligible probability. (The parameter ℓ is defined externally.) What makes the SDH assumption useful is that it implies the hardness of the following problem:

> On input two generators $g, g^{\omega} \in G_p$, and $\ell{-}1$ distinct pairs $(c_i, g^{1/(\omega+c_i)}) \in \mathbb{Z}_p^* {\times} G_p$, output an additional pair $(c, g^{1/(\omega+c)}) \in \mathbb{Z}_p^* {\times} G_p$ such that $c \neq c_i$ for all $i = 1, \ldots, \ell - 1$.

This argument was used by Boneh and Boyen [7] as the basis of their secure signature constructions. In particular, Boneh and Boyen's primordial "weakly secure signature" on a message c is nothing more than the group element $g^{1/(\omega+c)}$. Much of their paper is concerned with securing these signatures against *adaptive* chosen message attacks, but for our purposes this is unnecessary.

However, an inherent trait of the general notion of signature is that verification requires knowledge of the message. Since in our group signature the first-level "message" is the identity of the user, we would like to keep it as hidden as possible, since at the end of the day we need to blind it. To facilitate this task, we build a modified version of the Boneh-Boyen "weak signature" above that does not require knowledge of c in order to verify. It is based on the Hidden SDH assumption, a straightforward extension to the SDH assumption where the "message" c is not given in the clear.

The Hidden Strong Diffie-Hellman Problem. We first define the ℓ-HSDH problem as follows:

> On input three generators $g, h, g^{\omega} \in G_p$, and $\ell - 1$ distinct triples $(g^{1/(\omega+c_i)}, g^{c_i}, h^{c_i}) \in G_p^3$ where $c_i \in \mathbb{Z}_p$, output another such triple $(g^{1/(\omega+c)}, g^c, h^c) \in G_p^3$ distinct of all the others.

Observe that the well-formedness of a triple $(A, B, C) = (g^{1/(\omega+c)}, g^c, h^c)$ can be ascertained without knowing c by verifying that $e(A, g^\omega B) = e(g, g)$ and that $e(B, h) = e(C, g)$. In these verifications, the Diffie-Hellman relationship (g, h, g^c, h^c) serves as a discrete-log NIZK proof of knowledge of c. Notice that contrary to the SDH problem statement [7], here we allow c or some c_i to be zero.

We define the advantage of an HSDH adversary $\mathcal{A}$ as its probability of outputting a valid triple. The probability is taken over the random choice of instance and the random bits used by $\mathcal{A}$.

Definition 1. *We say that the ℓ-HSDH assumption holds in a family of prime order bilinear groups generated by $\mathcal{GG}$, if there is no PPT algorithm that, for sufficiently large $\lambda \in \mathbb{N}$, solves the HSDH problem in the bilinear group $(p, G_p, e) \leftarrow \mathcal{GG}(1^\lambda)$ with non-negligible probability. Here, ℓ may be either an explicit parameter to the assumption, or some polynomially bounded function of the security parameter λ.*

It is easy to see that for any $\ell \geq 1$, hardness of the ℓ-HSDH problem implies hardness of the ℓ-SDH problem in the same group, which itself requires the CDH problem to be hard in that group. To bolster our confidence in the new complexity assumption, we can prove an $\Omega(\sqrt{p/\ell})$ lower bound on the complexity of solving the HSDH problem in generic bilinear groups, provided that $\ell < \sqrt[3]{p}$. Notice that HSDH does not rely on the composite order n, so the generic group model can apply. The proof will appear in the full paper.

4 Anonymous Hierarchical Signatures

As our first step toward short group signatures, we build a hierarchical signature with the signer identity at the first level and the message being signed at the second level, such that the whole signature can be verified without revealing the identity.

In a hierarchical signature, a message is a tuple comprising several atomic message components. The crucial property is that a signature on a message $(m_1, \ldots, m_i)$, also acts as a restricted private key that enables the signing of any message extension $(m_1, \ldots, m_i, \ldots, m_j)$ of which the original message is a prefix. In some schemes, the hierarchy has a maximum depth d, in which case we must have $i \leq j \leq d$. Here, we shall only consider 2-level hierarchical signatures, in which the first level is concerned with user identities, and the second level with messages proper. Notice that 2-level hierarchical signatures and identity-based signatures are equivalent notions: the identity-based key is just a fancy name for a signature on a first-level atomic component.

We use the HSDH assumption to construct a short two-level hierarchical signature that can be verified without knowing the user identity at the first level. Our construction makes a hybrid of two schemes, one at each level.

First Level. At the first level, we devise a variant of the "primary" deterministic Boneh-Boyen signatures from [7, §3.2]. Recall that Boneh-Boyen signatures are constructed in two stages, beginning with a primary "weak" deterministic signature, which is subsequently hardened with a sprinkle of randomness. The primary signature is weaker for the reason that in the forgery game, the opponent must submit all the signing queries up front, rather than adaptively as in the full Boneh-Boyen signature.

In the context of group signatures, this up-front attack model is perfectly adequate for signatures on user identities, since, in group signatures, user identities are not subject to adaptive attacks. Indeed, since there are only polynomially users in a group, their identities can be assigned from a polynomially sized set of integers. Furthermore, these unique identifiers can all be selected in advance by the group manager, and assigned to the users as they enroll in the system.

We shall make one modification to the primary Boneh-Boyen signatures. The modification will allow them to be verifiable without knowledge of the user identity. This is where our new HSDH assumption will come into play.

Second Level. At the second level, where the actual messages are signed, we can work with any secure signature scheme that can be meshed into an upward hierarchy. Hierarchical identity-based encryption schemes with "adaptive-identity security" make good candidates, since we can turn them into signatures schemes that are existentially unforgeable against adaptive chosen message attacks. We shall use a signature based on Waters' IBE scheme [29] for this purpose.

4.1 Hybrid Scheme

Let thus λ be the security parameter. User identities will be modeled as integers taken from a (non-public) polynomially sized random set $\{s_1, \ldots, s_{2^k}\} \subset \mathbb{Z}_p$ where $k = O(\log(\lambda))$. For convenience, we use sequential identifiers $\mathsf{ID} = 1, \ldots, 2^k$ to index the hidden identities s_{ID}, which are kept secret. Messages will be taken as binary strings of fixed length $m = O(\lambda)$. In the description that follows, g is a generator of the prime order subgroup G_p; therefore all group elements in the basic hierarchical signature scheme will have prime order p in G and G_T.

Setup(1^λ)**:** To setup the system, first, secret integers $\alpha, \omega \in \mathbb{Z}_p$ are chosen at random, from which the values $\Omega = g^\omega$ and $A = e(g,g)^\alpha$ are calculated. Next, two integers $y, z' \in \mathbb{Z}_p$ and a vector $\boldsymbol{z} = (z_1, \ldots, z_m) \in \mathbb{Z}_p^m$ are selected at random. The public parameters and master key are

$$\mathsf{PP} = \Big(\ g,\ \Omega = g^\omega,\ u = g^y,\ v' = g^{z'},\ v_1 = g^{z_1},\ \ldots,\ v_m = g^{z_m},\ A = e(g,g)^\alpha\ \Big) \in G^{m+5} \times G_T$$

$$\mathsf{MK} = \Big(\ \omega,\ \ g^\alpha,\ \ s_1\ , \ldots,\ s_{2^k}\ \ \Big) \in \mathbb{Z}_p \times G \times \mathbb{Z}_p^{2^k}$$

The public parameters, PP, also implicitly include k, m, and a description of (p, G, G_T, e). The master key, MK, is assumed to contain the secret list of user identities, $\{s_1, \ldots, s_{2^k}\} \subset \mathbb{Z}_p$.

Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$**:** To create a private key for the identity s_{ID} associated with the user of index $1 \leq \mathsf{ID} \leq 2^k$, return

$$K_{\mathsf{ID}} = \Big(\ (g^{\alpha})^{\frac{1}{\omega + s_{\mathsf{ID}}}}, \ \ g^{s_{\mathsf{ID}}}, \ \ u^{s_{\mathsf{ID}}} \ \Big) \in G^3$$

Sign$(\mathsf{PP}, K_{\mathsf{ID}}, M)$**:** To sign a message represented as a bit string $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$, using a private key $K_{\mathsf{ID}} = (K_1, K_2, K_3) \in G^3$, select a random $s \in \mathbb{Z}_p$, and output

$$S = \Big(\ K_1, \ \ K_2, \ \ K_3 \cdot (v' \prod_{j=1}^{m} v_j^{\mu_j})^s, \ \ g^{-s} \ \Big) \in G^4$$

Verify(PP, M, σ)**:** To verify that a signature $S = (S_1, S_2, S_3, S_4) \in G^4$ is valid for a message $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$, check whether

$$e(\ S_1 \ , \ S_2\, \Omega\ ) \stackrel{?}{=} A \qquad \text{and} \qquad e(\ S_2 \ , \ u\ ) \stackrel{?}{=} e(\ S_3 \ , \ g\ ) \cdot e(\ S_4 \ , \ v' \prod_{j=1}^{m} v_j^{\mu_j}\ )$$

It the equality holds, output `valid`; otherwise, output `invalid`.
Notice that in this case we did not verify the signer's identity, ID, only the message, M. However, signatures remain linkable because S_2 and S_3 are invariant for the same user.

4.2 Existential Unforgeability

The hybrid scheme is existentially unforgeable against adaptive chosen message attacks, and is anonymous at the first level. We shall now state and prove the unforgeability property, which will be needed later on when building group signatures.

Theorem 1. *Consider an adversary $\mathcal{A}$ that existentially forges the hybrid two-level signature scheme in an adaptive chosen message attack. Assume that $\mathcal{A}$ makes no more that $\ell - 1 \ll p$ signature queries and produces a successful forgery with probability ϵ in time t. Then there exists an algorithm $\mathcal{B}$ that solves the ℓ-HSDH problem with probability $\tilde{\epsilon} \approx \epsilon/(4m\ell^2)$ in time $\tilde{t} \approx t$.*

The proof of this theorem uses a two-prong strategy, one for each level. At the first level, we give a reduction based on the ℓ-HSDH assumption, where $\ell = 2^k$ is the number of secret user identities in the master key list (or the number that we have actually used). At the second level, we construct a reduction from the CDH assumption in the bilinear group G_p, but since CDH is implied by HSDH, we get a single reduction from HSDH for both levels at once. All reductions are in the standard model.

Proof. The proof may be found in Appendix A.

5 Constant-Size Group Signatures

We now describe the actual group signature scheme, based on the hierarchical signature scheme above. It is obtained from by obfuscating the user identity, and replacing it by a NIZK proof of it being well formed. We also need to incorporate a tracing mechanism, which is achieved by using a trapdoor into the NIZK proof.

5.1 Related Schemes

The group signature we describe invites comparison with two earlier schemes that also feature compact signatures and provable security without random oracles. One of the earlier schemes is due to Boyen and Waters [11, 12], the other to Ateniese et al. [1].

The key difference with the earlier Boyen-Waters group signature scheme [11, 12], is that the earlier scheme relied on an all-purpose bit hiding technique due to Groth, Ostrovsky, and Sahai [22] to conceal the user identity. Unfortunately, each bit had to supply its own NIZK proof in the final signature, which resulted in a logarithmic-size group signature. The present scheme manages to give a single short proof for the entire identity at once. This makes the resulting signature much shorter, comprising only a small, constant number of group elements.

One of the main differences with the Ateniese et al. [1] scheme, is that the latter relied on very strong, interactive complexity assumptions in order to implement the corresponding NIZK proofs. The present scheme is simpler, and arguably rests on firmer ground.

5.2 Core Construction

The group signature scheme is described by the following algorithms.

Setup(1^λ)**:** The input is a security parameter in unary, 1^λ. Suppose we wish to support up to 2^k signers in the group, and sign messages in $\{0,1\}^m$, where $k = O(\lambda)$ and $m = O(\lambda)$.

The setup algorithm first chooses $n = pq$ where p and q are random primes of bit size $\lceil \log_2 p \rceil, \lceil \log_2 q \rceil = \Theta(\lambda) > k$. From this, it builds a cyclic bilinear group G of order n. Denote by G_p and G_q the cyclic subgroups of G of respective order p and q. The algorithm also selects a generator g of G and a generator h of G_q. Next, the algorithm picks two random exponents $\alpha, \omega \in \mathbb{Z}_n$, and defines $A = e(g,g)^\alpha \in G_T$ and $\Omega = g^\omega \in G$. Finally, it draws $m+2$ random generators, $u, v', v_1, \ldots, v_m \in G$.

The public information consists of the bilinear group, (n, G, G_T, e), and the public values,

$$\begin{aligned} \mathsf{PP} = \Big(\ g,\ h,\ u,\ v',\ v_1,\ \ldots,\ v_m,\ \Omega = g^\omega,\ A = e(g,g)^\alpha \ \Big) \\ \in G \times G_q \times G^{m+3} \times G_T \end{aligned}$$

The master enrollment key, MK, and the group manager's tracing key, TK, are, respectively,

$$\mathsf{MK} = \big(\, g^{\alpha},\ \omega \,\big) \ \in G \times \mathbb{Z}_n \qquad\qquad \mathsf{TK} = q \ \in \mathbb{Z}$$

Enroll$(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$**:** Suppose we wish to create a signing key for user ID, where $0 \leq \mathsf{ID} < 2^k < p$. Upon enrollment in the group, the user is assigned a secret unique value $s_{\mathsf{ID}} \in \mathbb{Z}_n$, to be later used for tracing purposes. This value must be chosen so that $\omega + s_{\mathsf{ID}}$ lies in $\mathbb{Z}_n^{\times}$, the multiplicative group modulo n. Based on the hidden identity s_{ID}, the signing key to be given to the user is constructed as,

$$K_{\mathsf{ID}} = (K_1, K_2, K_3) = \Big(\ (g^{\alpha})^{\frac{1}{\omega + s_{\mathsf{ID}}}},\ g^{s_{\mathsf{ID}}},\ u^{s_{\mathsf{ID}}}\ \Big) \ \in G^3$$

Here, K_1 is essentially a deterministic Boneh-Boyen signature on s_{ID}, which is not disclosed. Rather, K_2 and K_3 provide a NIZK proof of knowledge of s_{ID} by the issuing authority. There is also a supplemental constant exponent α that will matter at the second level. The newly enrolled user may verify that the key is well formed by checking that (cfr. Section 4),

$$e(K_1, K_2\, \Omega) \overset{?}{=} A \qquad \text{and} \qquad e(K_2, u) \overset{?}{=} e(K_3, g).$$

Sign$(\mathsf{PP}, \mathsf{ID}, K_{\mathsf{ID}}, M)$**:** To sign a message $M = (\mu_1 \dots \mu_m) \in \{0,1\}^m$, a user with a signing key K_{ID} proceeds as follows.

First, K_{ID} is used to create a two-level hybrid signature with the message M at the second level. To do so, the user chooses a random $s \in \mathbb{Z}_n$ and computes the (randomized but unblinded) hybrid signature,

$$\theta = (\theta_1, \theta_2, \theta_3, \theta_4) = \left(\ K_1,\ \ K_2,\ \ K_3 \cdot \Big(v' \prod_{i=1}^{m} v_i^{\mu_i}\Big)^{s},\ \ g^{-s}\ \right)$$

Notice that this initial signature satisfies the regular verification equations: $e(\theta_1, \theta_2\, \Omega) = A$, and $e(\theta_2, u) = e(\theta_3, g) \cdot e(\theta_4, v' \prod_{i=1}^{m} v_i^{\mu_i})$.

Next, θ must be turned into a blinded signature that is both verifiable and traceable, but remains unlinkable and anonymous to anyone who lacks the tracing key. To proceed, the signer picks four random exponents $t_1, t_2, t_3, t_4 \in \mathbb{Z}_n$ and sets,

$$\sigma_1 = \theta_1 \cdot h^{t_1}, \qquad \sigma_2 = \theta_2 \cdot h^{t_2}, \qquad \sigma_3 = \theta_3 \cdot h^{t_3}, \qquad \sigma_4 = \theta_4 \cdot h^{t_4}.$$

Additionally, it computes the two group elements,

$$\pi_1 = h^{t_1 t_2} \cdot (\theta_1)^{t_2} \cdot \big(\theta_2\, \Omega\big)^{t_1}, \qquad\qquad \pi_2 = u^{t_2} \cdot g^{-t_3} \cdot \Big(v' \prod_{i=1}^{m} v_i^{\mu_i}\Big)^{t_4}.$$

The final signature is output as:

$$\sigma = \big(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2\big) \ \in G^6.$$

Verify(PP, M, σ)**:** To validate a group signature σ on a message M, the verifier first calculates,

$$T_1 = A^{-1} \cdot e(\sigma_1, \sigma_2\, \Omega), \qquad T_2 = e(\sigma_2, u) \cdot e(\sigma_3, g)^{-1} \cdot e(\sigma_4, v' \prod_{i=1}^{m} v_i^{\mu_i})^{-1}.$$

Then it checks whether,

$$T_1 \stackrel{?}{=} e(h, \pi_1), \qquad T_2 \stackrel{?}{=} e(h, \pi_2).$$

If both equalities hold, the verifier outputs `valid`; otherwise, it outputs `invalid`.

These tests show that $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is a valid 2-level hybrid signature once the random blinding factors are removed; the extra elements (π_1, π_2) serve to convince the verifier that the blinding factors were affixed correctly.

Trace$(\mathsf{PP}, \mathsf{TK}, \sigma)$**:** Let $\sigma = (\ldots, \sigma_2, \ldots)$ be a signature assumed to pass the verification test for some message M, which will not be needed here. To recover the identity of the signer, the tracing authority first calculates $(\sigma_2)^q$ using the tracing key TK. Then, for each auspicious identity ID_i, it tests whether,

$$(\sigma_2)^q \stackrel{?}{=} (g^{s_{\mathsf{ID}_i}})^q.$$

The tracer outputs the recovered identity, $\mathsf{ID} = \mathsf{ID}_i$, upon satisfaction of the above equation.

Remark that tracing can be done in constant time — the time to compute $(\sigma_2)^q$ — with the help of a lookup table of associations $(g^{s_{\mathsf{ID}_i}})^q \mapsto \mathsf{ID}_i$ for all users in the group. Since the value $(g^{s_{\mathsf{ID}_i}})^q$ can be calculated once and for all for each user ID_i, for instance upon a user's initial enrollment, the amortized cost of tracing is indeed essentially constant.

5.3 Security Properties

We now state the security properties of our constant-size group signature scheme.

Full Anonymity (under CPA attack) We prove the security of our group signature scheme in the anonymity game against chosen plaintext attacks. First, we show that an adversary cannot tell whether h is a random generator of G_q or G. Next, we show that if h is chosen from G then the identity of a signer is perfectly hidden, in the information theoretic sense.

Theorem 2. *Suppose no t-time adversary can solve the subgroup decision problem with advantage at least ϵ_{sd}. Then for every t'-time adversary $\mathcal{A}$ where $t' \approx t$ we have that $\mathrm{Adv}_{\mathcal{A}} < 2\,\epsilon_{\mathrm{sd}}$.*

Proof. We use a game switching argument where $\varGamma_0$ is the real group signature anonymity game, and $\varGamma_1$ is a game in which the public parameters are the same as in the original game except that h is chosen randomly from G instead of G_q. We denote the adversary's advantage in the original game by $\mathrm{Adv}_{\mathcal{A}}$, and in the modified game by $\mathrm{Adv}_{\mathcal{A},\Gamma_1}$.

First, in Lemma 1, we show that the two games are essentially indistinguishable, unless the Decision Subgroup assumption is easy. Second, in lemma 2, we use an information-theoretic argument to prove that in the game Γ_1 the adversary's advantage must be zero. The theorem follows from these results.

Lemma 1. *For all t'-time adversaries as above,* $\mathrm{Adv}_{\mathcal{A}} - \mathrm{Adv}_{\mathcal{A},\Gamma_1} < 2\,\epsilon_{\mathrm{sd}}$.

Lemma 2. *For any algorithm $\mathcal{A}$, we have that* $\mathrm{Adv}_{\mathcal{A},\Gamma_1} = 0$.

Proof (Proofs.). The proofs of these two lemmas are given in Appendix B.1.

Full Traceability We reduce the full traceability of the group signature scheme to the existential unforgeability of the underlying hybrid signature construction of Section 4.

Theorem 3. *If there exists a (t, ϵ) adversary for the full traceability game against the group signature scheme, then there exists a $(\tilde{t}, \epsilon)$ adaptive chosen message existential unforgeability adversary against the two-level hybrid signature scheme, where $t \approx \tilde{t}$.*

Proof. We prove this theorem in Appendix B.2.

6 CCA-Security

In the introduction we stated that the two primary drawbacks of the Boyen-Waters [12] scheme are that the signature grew logarithmically with the number of signers and that the scheme was not CCA secure. In this work we addressed the first limitation, but left the second one open. Here we explain some of the challenges in achieving CCA security while using the subgroup paradigm for proofs.

In both this paper and the Boneh-Waters scheme the authority uses knowledge of the factorization of the group order in order to trace. In order to achieve CCA security we will clearly need to take a different approach since all known CCA proof techniques depend upon a simulation knowing partial decryption information (e.g. consider the two key paradigm of Dolev, Dwork and Naor [19]).

One tempting direction is to provably encrypt (in a simulation sound manner) the identity of the signer in one of the recent bilinear map based CCA-secure cryptosystems derived from the techniques of Canetti, Halevi, and Katz [17] . Then we could allow the tracer have the decryption key for this system, but not know the group's factorization. However, there is one large problem with this technique. The subgroup-based NIZK techniques only prove soundness in one subgroup. It is easy to see that a corrupt signer can provably encrypt his identity and then randomize the encryption in one subgroup. Since the decryption authority will not know the factorization, his view of the identity will be indistinguishable from random. Therefore, it seems more complex techniques are necessary to achieve CCA-security will using subgroup based proofs. This might also be an argument for basing future group signature schemes on the decisional-linear [8] assumption proofs [21].

References

1. Giuseppe Ateniese, Jan Camenisch, Susan Hohenberger, and Breno de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org/`.
2. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proceedings of Crypto 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–70. Springer-Verlag, 2000.
3. Giuseppe Ateniese, Dawn Song, and Gene Tsudik. Quasi-efficient revocation of group signatures. In *Proceedings of Financial Cryptography 2002*, 2002.
4. Giuseppe Ateniese and Gene Tsudik. Some open issues and directions in group signatures. In *Proceedings of Financial Cryptography 1999*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1999.
5. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–29. Springer-Verlag, 2003.
6. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *Proceedings of CT-RSA 2005*, Lecture Notes in Computer Science, pages 136–153. Springer-Verlag, 2005.
7. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer-Verlag, 2004.
8. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer-Verlag, 2004.
9. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proceedings of TCC 2005*, Lecture Notes in Computer Science. Springer-Verlag, 2005.
10. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of ACM CCS 2004*, pages 168–77. ACM Press, 2004.
11. Xavier Boyen and Brent Waters. Compact group signatures without random oracles. Cryptology ePrint Archive, Report 2005/381, 2005. `http://eprint.iacr.org/`.
12. Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444. Springer-Verlag, 2006.
13. Jan Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology—EUROCRYPT 1997*, Lecture Notes in Computer Science, pages 465–479. Springer-Verlag, 1997.
14. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *Proceedings of SCN 2004*, pages 120–133, 2004.

15. Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology—CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer-Verlag, 2002.
16. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.
17. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*. Springer-Verlag, 2004.
18. David Chaum and Eugène van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 257–65. Springer-Verlag, 1991.
19. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552, 1991.
20. Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In *ASIACRYPT*, pages 444–459, 2006.
21. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive Zaps and new techniques for NIZK. In *Advances in Cryptology—CRYPTO 2006*, Lecture Notes in Computer Science. Springer-Verlag, 2006.
22. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *Advances in Cryptology—EUROCRYPT 2006*, Lecture Notes in Computer Science. Springer-Verlag, 2006. To appear.
23. Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4), 2003.
24. Aggelos Kiayias and Moti Yung. Extracting group signatures from traitor tracing schemes. In *Advances in Cryptology—EUROCRYPT 2003*, Lecture Notes in Computer Science, pages 630–48. Springer-Verlag, 2003.
25. Aggelos Kiayias and Moti Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. `http://eprint.iacr.org/`.
26. Aggelos Kiayias and Moti Yung. Group signatures with efficient concurrent join. In *Advances in Cryptology—EUROCRYPT 2005*, Lecture Notes in Computer Science, pages 198–214. Springer-Verlag, 2005.
27. Moni Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology—CRYPTO 2003*, Lecture Notes in Computer Science, pages 96–109. Springer-Verlag, 2003.
28. Dawn Xiaodong Song. Practical forward secure group signature schemes. In *ACM Conference on Computer and Communications Security—CCS 2001*, pages 225–234, 2001.
29. Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

A Security of the Anonymous Hybrid Signature

We reduce the HSDH problem to the existential forgery of the signature scheme of Section 4.

Proof (Proof of Theorem 1.). The adversary may make requests for signatures at the first and second level. Since these two are constructed differently, we need to consider two types of forgeries. Let ID_i and (ID_i, M_i) be the first- and second-level queries made by $\mathcal{A}$, and let (ID^*, M^*) be the second-level target of $\mathcal{A}$'s eventual forgery (clearly, if $\mathcal{A}$ chooses a first-level target, it is easy to turn it into a second-level one). In a type-1 forgery, ID^* is distinct from all ID_i. In a type-2 forgery, (ID^*, M^*) is distinct from all (ID_i, M_i) though $\mathsf{ID}^* = \mathsf{ID}_i$ for some i. We first give reductions for both cases in lemmas 3 and 4; then we shall conclude the proof of the theorem.

Lemma 3. *If there exists an algorithm $\mathcal{A}$ that makes $\ell - 1$ signature queries and outputs a type-1 forgery with probability ϵ in time t, then there exists an algorithm $\mathcal{B}$ that solves the ℓ-HSDH problem with probability $\tilde{\epsilon}_1 \geq \epsilon - (\ell - 1)/p \approx \epsilon$ in time $\tilde{t}_1 \approx t$.*

Proof. The simulator $\mathcal{B}$ is given an instance $(g, u, g^\omega, (A_i = g^{1/(\omega+c_i)}, B_i = g^{c_i}, C_i = u^{c_i})_{i=1,\ldots,\ell-1})$ of the HSDH problem, for some undisclosed ω. To prepare the simulation, $\mathcal{B}$ first selects random exponents $\alpha, z', z_1, \ldots, z_m \in \mathbb{Z}_p$. It gives $\mathcal{A}$ the public key, $\mathsf{PP} = (g, \Omega = g^\omega, u, v' = g^{z'}, v_1 = g^{z_1}, \ldots, v_m = g^{z_m}, A = e(g, g)^\alpha)$. $\mathcal{B}$ also maintains a table, initially empty, of mappings from identities ID to indices $i \in \{1, \ldots, \ell - 1\}$.

To answer a signature query on ID or (ID, M), $\mathcal{B}$ proceeds as follows. Suppose this is the j-th query, for $0 < j < \ell$. If ID has been seen in a previous query, let i be the index previously associated with it. If ID is new, $\mathcal{B}$ associates to ID the index $i = j$. In all cases, $\mathcal{B}$ lets $K_{\mathsf{ID}} = (A_i^\alpha, B_i, C_i)$. If the query was first-level (on ID), it simply responds with K_{ID}. If instead the query was second-level, on (ID, M), the response is $S \leftarrow \mathit{Sign}(\mathsf{PP}, K_{\mathsf{ID}}, M)$.

After at most $\ell - 1$ total queries, $\mathcal{A}$ outputs a forgery $S^* = (S_1, S_2, S_3, S_4)$ on (ID^*, M^*), where $S_2 = g^{c^*}$ for some $c^* \in \mathbb{Z}_p$ chosen by the forger. In a type-1 forgery, ID^* never appeared in any query, which means that $S_2 = g^{s_{\mathsf{ID}^*}}$ must be distinct from all values $B_i = g^{c_i}$ used by the simulator; a random collision with an unused B_i is possible, but only with negligible probability $(\ell - 1)/p \approx 0$. The simulator outputs its own answer as, $(S_1^{1/\alpha},\ S_2,\ S_3 \cdot S_4^{z' + \sum_{j=1}^m z_j \mu_j})$.

Lemma 4. *If there exists an algorithm $\mathcal{A}$ that makes $\ell - 1$ signature queries and outputs a type-2 forgery with probability ϵ in time t, then there exists an algorithm $\mathcal{B}$ that solves the ℓ-HSDH problem with probability $\tilde{\epsilon}_2 = \epsilon/(4m\ell^2)$ in time $\tilde{t}_2 \approx t$.*

Proof (Proof of Lemma 4.). The simulator $\mathcal{B}$ is given an instance $(g, u, g^\omega, (A_i = g^{1/(\omega+c_i)}, B_i = g^{c_i}, C_i = u^{c_i})_{i=1,\ldots,\ell-1})$ of the HSDH problem. To set up the simulation, $\mathcal{B}$ first guesses the index $i^* \in \{1, \ldots, \ell - 1\}$ of the identity that the adversary will choose to attack. It also chooses a random $k \in \{0, \ldots, m\}$. Next, the simulator chooses random numbers $x', x_1, \ldots, x_m$ independently and uniformly at random in the interval $\{0, \ldots, 2\ell - 1\}$. It also chooses random numbers $\alpha, z', z_1, \ldots, z_m \in \mathbb{Z}_p$. Last, it selects $t \in \mathbb{Z}_p$ and lets $f = \Omega^{-1} g^t$. The simulator gives to $\mathcal{A}$ the public key,

$$\mathsf{PP} = \Big(\ g,\ \Omega = g^{\omega},\ u,\ v' = f^{x'-2k\ell} g^{z'},\ v_1 = f^{x_1} g^{z_1},\ \ldots,\ v_m = f^{x_m} g^{z_m},\ A = e(g,g)^{\alpha}\ \Big).$$

It also maintains a table, initially empty, of mappings from identities ID to indices $i \in \{1, \ldots, \ell - 1\}$.

To answer signature queries of the form ID or (ID, M), the first task is to associate an index i to the identity ID. If ID appeared in a previous query, the old index is used. Otherwise, a new index is assigned sequentially, starting from 1. Note that $1 \leq i \leq \ell - 1$. The rest depends of the index.

Whenever $i \neq i^*$, the simulator lets $K_{\mathsf{ID}} = (A_i^{\alpha}, B_i, C_i)$ from the HSDH instance. If the query was first-level, $\mathcal{B}$ returns K_{ID} as the answer. If the query was second-level on a message M, then $\mathcal{B}$ responds with the signature given by running $\mathit{Sign}(\mathsf{PP}, K_{\mathsf{ID}}, M)$ as in the real protocol.

Whenever $i = i^*$, the simulator eschews (A_i^{α}, B_i, C_i); instead, it pretends to build its answer(s) on the private key partially simulated by $(g^{\alpha/t}, f, \ldots)$, which corresponds to $(g^{\alpha/(\omega+s^*)}, g^{s^*}, u^{s^*})$ for the unknown integer $s^* = t - \omega$. Concretely, if the query is first-level, then $\mathcal{B}$ cannot proceed and aborts. For a second-level query on $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$, define $F = -2k\ell + x' + \sum_{j=1}^m x_j \mu_j$ and $J = z' + \sum_{j=1}^m z_j \mu_j$. In case $F \equiv 0 \pmod{p}$, then $\mathcal{B}$ aborts the simulation. Otherwise, $\mathcal{B}$ picks a random $r \in \mathbb{Z}_p$, and returns the signature,

$$S = (S_1, S_2, S_3, S_4) = \Big(\ g^{\alpha/t},\ \ \Omega^{-1} g^{t},\ \ u^{-J/F} (v' \prod_{j=1}^{m} v_j^{\mu_j})^{r},\ \ u^{1/F} g^{-r}\ \Big).$$

For $\tilde{r} = r - \mathrm{dlog}_g(u)/F$ and $s^* = t - \omega$, we find that $S_3 = u^{-J/F}(v' \prod_{j=1}^m v_j^{\mu_j})^r = u^{-J/F}(f^F g^J)^r = u^{-J/F}(f^F g^J)^{\tilde{r}} f^{\mathrm{dlog}_g(u)} u^{J/F} = (\Omega^{-1} g^t)^{\mathrm{dlog}_g(u)} (v' \prod_{j=1}^m v_j^{\mu_j})^{\tilde{r}} = u^{s^*}(v' \prod_{j=1}^m v_j^{\mu_j})^{\tilde{r}}$, and, similarly, that $S_4 = u^{1/F} g^{-r} = u^{1/F} g^{-\tilde{r}} u^{-1/F} = g^{-\tilde{r}}$. This leaves us with a correctly distributed signature,

$$S = (\ \ g^{\alpha/(\omega+s^*)},\ \ g^{s^*},\ \ u^{s^*} (v' \prod_{j=1}^{m} v_j^{\mu_j})^{\tilde{r}},\ \ g^{-\tilde{r}}\ \ ).$$

Eventually, the adversary outputs a valid type-2 forgery $S^* = (S_1^*, S_2^*, S_3^*, S_4^*)$ on a message $M^* = (\mu_1^* \ldots \mu_m^*) \in \{0,1\}^m$. If $S_2^* \neq f$, the simulator must abort, having incorrectly guessed which identity would be targeted. Otherwise, let $F^* = -2k\ell + x' + \sum_{j=1}^m x_j \mu_j^*$ and $J^* = z' + \sum_{j=1}^m z_j \mu_j^*$. If $F^* \not\equiv 0 \pmod{p}$, the simulator also aborts, having received a useless forgery. Otherwise, the forgery must be of the form,

$$S^* = (\ \ g^{\alpha/(\omega+s^*)},\ \ g^{s^*},\ \ u^{s^*} (f^0 g^{J^*})^{r},\ \ g^{-r}\ \ ).$$

To solve the HSDH instance, our algorithm $\mathcal{B}$ outputs the triple $(S_1^{1/\alpha},\ S_2,\ S_3 \cdot S_4^{J^*})$.

To conclude the proof, we must find the probability that $\mathcal{B}$ completes the simulation without aborting. First, the guess on i^* must be correct, which has probability $1/(\ell - 1)$. Second, for each query with $i = i^*$, the conditional probability that $F \not\equiv 0 \pmod{p}$ given that it has made it so far is at least $1 - 1/(2\ell)$; and since there are less than ℓ such queries, the total probability of not aborting through all of them is greater than $1/2$. Last, the conditional probability that $F^* \equiv 0 \pmod{p}$ upon reaching the challenge stage is at least $1/(2m\ell)$. Since $\mathcal{A}$ succeeds with probability ϵ when the simulation is not aborted, we conclude that $\mathcal{B}$ succeeds with probability $\tilde{\epsilon}_2 \geq \epsilon/(4m\ell(\ell - 1)) \geq \epsilon/(4m\ell^2)$.

We can now finish our proof of Theorem 1.

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the HSDH algorithms shown in the two lemmas above. To prove the theorem, it suffices to let $\mathcal{B}$ run $\mathcal{B}_1$ or $\mathcal{B}_2$ with probability $1/(4m\ell^2 + 1)$ and $1 - 1/(4m\ell^2 + 1)$ respectively. Since both simulations are perfect, the adversary cannot distinguish either simulator from a real attack environment, and outputs either a type-1 or type-2 forgery with the same probability in each case. It follows that $\mathcal{B}$ solves ℓ-HSDH with probability $\tilde{\epsilon} \approx \epsilon/(4m\ell^2 + 1)$ in time $\tilde{t} = \max(\tilde{t}_1, \tilde{t}_2) \approx t$.

B Security of the Constant-Size Group Signature

We prove the security properties of the constant-size group signature scheme of Section 5.

B.1 Full Anonymity (under CPA attack)

Proof (Proof of Lemma 1.). Consider an algorithm $\mathcal{B}$ that plays the subgroup decision problem. Upon receiving a subgroup decision challenge (n, G, G_T, e, w) the algorithm $\mathcal{B}$ first creates public parameters for the group signature scheme by setting $h = w$ and then choosing the remaining public parameters exactly as in the group signature scheme. It then sends the public information to $\mathcal{A}$ and plays the anonymity game with it. If w is randomly chosen from G_q then the game being played is the normal anonymity game; otherwise, if w is chosen randomly from G, then the game being played is a different game we call $\mathit{\Gamma}_1$. In either case, the algorithm $\mathcal{B}$ will be able to answer all queries—namely, issue private signing keys for, and sign any message on behalf of, any user—, since it knows the master key.

At some point the adversary will choose a message M and two identities ID_1 and ID_2 it wishes to be challenged upon, under the usual constraints that it had not previously made a signing key query on ID_x or a signature query on (ID_x, M). The simulator $\mathcal{B}$ will create the requisite challenge signature on M, and $\mathcal{A}$ will guess the identity of the signer. If $\mathcal{A}$ answers correctly, then $\mathcal{B}$ outputs $b = 1$, to signify that $w \in G_q$; otherwise it outputs $b = 0$, to signify that $w \in G$.

Denote by $\mathrm{Adv}_{\mathcal{B}}$ the advantage of the simulator $\mathcal{B}$ in the subgroup decision game. As we know that $\Pr[w \in G] = \Pr[w \in G_q] = \frac{1}{2}$, we deduce that,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}} - \mathrm{Adv}_{\mathcal{A},\mathit{\Gamma}_1} &= \Pr[b = 1 | w \in G_q] - \Pr[b = 1 | w \in G] \\ &= 2\Pr[b = 1, w \in G_q] - 2\Pr[b = 1, w \in G] = 2\,\mathrm{Adv}_{\mathcal{B}} < 2\,\epsilon_{\mathrm{sd}},\end{aligned}$$

since by our hardness assumption $\mathrm{Adv}_{\mathcal{B}}$ must be lesser than ϵ_{sd}, given that $\mathcal{B}$ runs in time $t \approx t'$.

Proof (Proof of Lemma 2.). We show that when h is chosen uniformly at random from G, instead of G_q in the real scheme, then the challenge signature is statistically independent of the challenge signer identity, ID, and thus s_{ID}, in the adversary's view. First, observe that only the challenge signature could possibly be dependent on ID, in the sense that all queries happen before the adversary announces ID_1 and ID_2 between which ID is to be chosen. However, since the tracing value s_{ID} may have been used to answer previous signing queries on (ID_1, M) or (ID_2, M), we have to show that the challenge signature is statistically independent of s_{ID}.

To proceed, we consider what a computationally unbounded adversary might deduce from the challenge signature,

$$\sigma = \big(\ \sigma_1, \sigma_2, \sigma_3, \sigma_4,\ \pi_1, \pi_2\ \big).$$

A priori, all components of σ except σ_4 depend on s_{ID} and could thus reveal it to an unbounded adversary. In addition we must assume that the following discrete logs are known to the computationally unbounded adversary,

$$y = \mathrm{dlog}_g(u), \quad \eta = \mathrm{dlog}_g(h), \quad \alpha, \quad \omega, \quad \mathrm{dlog}_g(v'), \quad \forall i : \mathrm{dlog}_g(v_i)$$

We define $V = v' \prod_{i=1}^{m} v_i^{\mu_i}$, which is also a known quantity given the message M.

To show that σ reveals no information about the value s_{ID}, we shall prove that σ is compatible with any hypothesis $\tilde{s}_{\mathsf{ID}}$ that the adversary might make about it. Observe that any value $\tilde{s}_{\mathsf{ID}}$ such that $\omega + \tilde{s}_{\mathsf{ID}} \in \mathbb{Z}_n^\times$ is *a priori* possible for s_{ID}, before the adversary is given σ. We need to show that the same set remains possible *a posteriori*, conditionally on σ. We proceed in three steps.

1. The first step is to note that the four values $\sigma_1, \ldots, \sigma_4$ by themselves reveal nothing about s_{ID}, as they are perfectly blinded by the four uniform and independent random factors $h^{t_1}, \ldots, h^{t_4} \in G$. In particular, regardless of the hypothesis $\tilde{s}_{\mathsf{ID}}$ contemplated by the adversary, there exists exactly one assignment to t_1 and t_2 that explains the values of σ_1 and σ_2 in the challenge signature for that hypothesis. (Note that the values of t_3 and t_4 depend on an additional hypothesis by the adversary about the value of s.)
2. For the second step, we show that the value of π_2 reveals t_4, and thus also s and t_3, but contains no additional information. To see this, notice that π_2 can be expressed as a linear combination of σ_2, σ_3, and V, whose coefficients are all constant except for t_4. To wit, we have,

   $$(\pi_2)^\eta = (u^{t_2} g^{-t_3} V^{t_4})^\eta = (h^{t_2})^y (h^{t_3})^{-1} V^{t_4 \eta} = (g^{s_{\mathsf{ID}}} h^{t_2})^y (u^{s_{\mathsf{ID}}} h^{t_3})^{-1} V^{t_4 \eta} = (\sigma_2)^y (\sigma_3)^{-1} V^{t_4 \eta}.$$

   Since π_2, σ_2, and σ_3 are given, V is implied, and η and y are constant given the system parameters, the only variable in the above equation is t_4. The equation always has a solution in the non-degenerate case where $V^\eta \neq 1$.

With t_4 thus determined, s follows (from σ_4), and so does t_3 (from σ_3). Since all of the foregoing holds regardless of the adversary's hypothesis, $\tilde{s}_{\mathsf{ID}}$, we deduce that none of the candidate values for s_{ID} can be ruled out, so far.

3. In the third step, we show that the remaining component, π_1, brings no new information to the adversary, which will conclude this argument. Let s^*_{ID} be the actual value of the hidden identifier used to create the challenge signature (recall that $\omega + s^*_{\mathsf{ID}} \in \mathbb{Z}_n^\times$); similarly, let t^*_1 and t^*_2 be the actual randomization exponents used by the challenger. Recall that $\sigma_1 = g^{\alpha/(\omega+s^*_{\mathsf{ID}})}\, h^{t^*_1}$ and $\sigma_2 = g^{s_{\mathsf{ID}}}\, h^{t^*_2}$. Thus, π_1 can be written,

$$\pi_1 = h^{t^*_1 t^*_2} \left(g^{\omega + s^*_{\mathsf{ID}}}\right)^{t^*_1} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}}\right)^{t^*_2}.$$

It suffices to show that the value of π_1 is consistent with the rest of σ in the adversary's view, and in particular that it does not contradict that view regardless of the hypothesis on s_{ID} that the adversary may have been entertaining. Again, let $\tilde{s}_{\mathsf{ID}}$ be the adversary's hypothesis, and let $\tilde{t}_1$ and $\tilde{t}_2$ be the values of t_1 and t_2 that are consistent with it (per our earlier argument in Steps 1 and 2). For notational convenience, we define,

$$\xi = \frac{\omega + s^*_{\mathsf{ID}}}{\omega + \tilde{s}_{\mathsf{ID}}} \pmod{n}.$$

By expanding the expression of σ_1 per the challenger's and the adversary's views, we find that,

$$\sigma_1 = g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}}\, h^{t^*_1} = g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}} + \eta t^*_1}, \qquad \text{and} \qquad \sigma_1 = g^{\frac{\alpha}{\omega + \tilde{s}_{\mathsf{ID}}}}\, h^{\tilde{t}_1} = g^{\frac{\alpha\xi}{\omega + s^*_{\mathsf{ID}}} + \eta \tilde{t}_1},$$

and thus,

$$\tilde{t}_1 = t^*_1 + \frac{\alpha\,(1-\xi)}{\eta\,(\omega + s^*_{\mathsf{ID}})} \pmod{n}.$$

By applying an analogous argument on the expression of the product $\varOmega\, \sigma_2$, we also find that,

$$\varOmega\, \sigma_2 = g^{\omega + s^*_{\mathsf{ID}}}\, h^{t^*_2} = g^{\omega + s^*_{\mathsf{ID}} + \eta t^*_2}, \qquad \text{and} \qquad \varOmega\, \sigma_2 = g^{\omega + \tilde{s}_{\mathsf{ID}}}\, h^{\tilde{t}_2} = g^{\frac{\omega + s^*_{\mathsf{ID}}}{\xi} + \eta \tilde{t}_2},$$

and thus,

$$\tilde{t}_2 = t^*_2 + \frac{\xi - 1}{\xi} \cdot \frac{\omega + s^*_{\mathsf{ID}}}{\eta} \pmod{n}.$$

Now, we need to show that the given value of π_1 is equal to what the adversary would expect based on its hypothesis. Taking the adversary's view, we work out the value of π_1 that it expects to be,

$$\begin{aligned}
\tilde{\pi}_1 &= h^{\tilde{t}_1 \tilde{t}_2} \left(g^{\omega + \tilde{s}_{\mathsf{ID}}}\right)^{\tilde{t}_1} \left(g^{\frac{\alpha}{\omega + \tilde{s}_{\mathsf{ID}}}}\right)^{\tilde{t}_2} \\
&= \left(g^{\omega + \tilde{s}_{\mathsf{ID}}}\right)^{\tilde{t}_1} \left(g^{\frac{\alpha \xi}{\omega + s^*_{\mathsf{ID}}} + \eta \tilde{t}_1}\right)^{\tilde{t}_2} \\
&= \left(g^{\omega + \tilde{s}_{\mathsf{ID}}}\right)^{\tilde{t}_1} \left(g^{\frac{\alpha \xi}{\omega + s^*_{\mathsf{ID}}} + \eta t^*_1 + \frac{\alpha\,(1-\xi)}{\omega + s^*_{\mathsf{ID}}}}\right)^{\tilde{t}_2} \\
&= \left(g^{\omega + \tilde{s}_{\mathsf{ID}}}\right)^{\tilde{t}_1} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}} + \eta t^*_1}\right)^{\tilde{t}_2} \\
&= \left(g^{\omega + \tilde{s}_{\mathsf{ID}}}\right)^{\tilde{t}_1} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}} h^{t^*_1}\right)^{t^*_2} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}} + \eta t^*_1}\right)^{\frac{\xi - 1}{\xi} \cdot \frac{\omega + s^*_{\mathsf{ID}}}{\eta}} \\
&= \left(g^{\frac{\omega + s^*_{\mathsf{ID}}}{\xi}}\right)^{t^*_1} g^{\frac{\alpha\,(1-\xi)}{\xi \eta}} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}} h^{t^*_1}\right)^{t^*_2} g^{\frac{\alpha\,(\xi - 1)}{\xi \eta}} \left(g^{t^*_1}\right)^{\frac{\xi - 1}{\xi} \cdot (\omega + s^*_{\mathsf{ID}})} \\
&= \left(g^{\omega + s^*_{\mathsf{ID}}}\right)^{t^*_1} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}} h^{t^*_1}\right)^{t^*_2} \\
&= h^{t^*_1 t^*_2} \left(g^{\omega + s^*_{\mathsf{ID}}}\right)^{t^*_1} \left(g^{\frac{\alpha}{\omega + s^*_{\mathsf{ID}}}}\right)^{t^*_2} = \pi_1 .
\end{aligned}$$

The value, $\tilde{\pi}_1$, expected by the adversary is seen to be equal to the actual value of π_1, regardless of the adversary's initial hypothesis s_{ID}. It follows that π_1 does not help the adversary to remove the uncertainty about s_{ID}.

The foregoing shows shows that the challenge signature, σ, is statistically independent of s_{ID} and thus ID, and hence that the advantage of any adversary in the anonymity game $\mathit{\Gamma}_1$ must necessarily be zero.

B.2 Full Traceability

Proof (Proof of Theorem 3.). Suppose there exists an algorithm $\mathcal{A}$ that is successful in the tracing game of our group signature scheme with advantage ϵ. Then we can create a simulator $\mathcal{B}$ that existentially forges signatures in an adaptive chosen message attack against the two-level signature scheme, with advantage ϵ.

The simulator will be given the factorization $n = pq$ of the group order $|G| = n$. As usual, denote by G_p and G_q the subgroups of G of respective order p and q, and by analogy let G_{Tp} and G_{Tq} be the subgroups of G_T of order p and q. The simulator begins by receiving from its challenger the public parameters of the signature game, all in subgroups of order p,

$$\tilde{\mathsf{PP}} = \Big(\ \tilde{g}\ ,\ \tilde{\mathit{\Omega}} = \tilde{g}^{\omega},\ \tilde{u} = \tilde{g}^{y},\ \tilde{v}' = \tilde{g}^{z'},\ \tilde{v}_1 = \tilde{g}^{z_1}, \ldots,\ \tilde{v}_m = \tilde{g}^{z_m},\ \tilde{A} = e(\tilde{g}, \tilde{g})^{\alpha}\ \Big) \in G_p^{m+4} \times G_{Tp} .$$

The simulator then picks random generators $(h, f, \gamma, \nu', \nu_1, \ldots, \nu_m) \in G_q^{m+4}$ and two random exponent $\beta, \psi \in \mathbb{Z}_q$. The simulator publishes the group public parameters as,

$$\mathsf{PP} = \Big(\ g = \tilde{g}\, f\ ,\ h\ ,\ u = \tilde{u}\, \gamma\ ,\ v' = \tilde{v}'\, \nu'\ ,\ v_1 = \tilde{v}_1\, \nu_1\ , \ldots,\ v_m = \tilde{v}_k\, \nu_m\ ,\ \mathit{\Omega} = \tilde{\mathit{\Omega}} \cdot f^{\psi}\ ,\ A = \tilde{A} \cdot e(f, f)^{\beta}\ \Big) .$$

The distribution of the public key is the same is in the real scheme. The simulator also gives the tracing key to the adversary,

$$\mathsf{TK} = q.$$

Suppose the adversary asks for the private key of user ID. To answer the query, the simulator first asks the challenger for a first-level signature on message ID, and receives back $\tilde{K}_{\mathsf{ID}} = (\tilde{K}_1, \tilde{K}_2, \tilde{K}_3) \in G_p^3$. Next, the simulator internally associates a persistent random $r_{\mathsf{ID}} \in \mathbb{Z}_q$ to ID, recalling the value previously associated to ID from storage as needed. It then creates the requested key as,

$$K_{\mathsf{ID}} = \Big(\; K_1 = \tilde{K}_1 \cdot (f^{\beta})^{\frac{1}{\psi + r_{\mathsf{ID}}}} \; , \;\; K_2 = \tilde{K}_2 \cdot f^{r_{\mathsf{ID}}} \; , \;\; K_3 = \tilde{K}_3 \cdot \gamma^{r_{\mathsf{ID}}} \; \Big).$$

This is a well formed key in our scheme. Indeed, since $\tilde{g} \in G_p$ and $h \in G_q$, it follows that $e(\tilde{g}, h) = 1$ in G_T, and hence,

$$e(K_1, K_2\, \Omega) = e(\tilde{g}^{\frac{\alpha}{\omega + s_{\mathsf{ID}}}} f^{\frac{\beta}{\psi + r_{\mathsf{ID}}}}, \tilde{g}^{\omega + s_{\mathsf{ID}}} f^{\psi + r_{\mathsf{ID}}}) = e(\tilde{g}, \tilde{g})^{\alpha}\, e(f, f)^{\beta} = A,$$

and similarly, $e(K_2, u) = e(K_3, g)$.

Suppose the simulator is asked for a signature on message $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$ from user ID. The simulator requests a two-level signature on (ID, M) and gets back a signature $S = (S_1, S_2, S_3, S_4) \in G_p^4$. Next, the simulator creates or recalls the persistent random value $r_{\mathsf{ID}} \in \mathbb{Z}_q$ associated to ID, as described above. It also chooses an ephemeral random exponent $r_0 \in \mathbb{Z}_q$. It then creates an unblinded signature, $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ as,

$$\theta = \Big(\; \theta_1 = S_1 \cdot (f^{\beta})^{\frac{1}{\psi + r_{\mathsf{ID}}}} \; , \;\; \theta_2 = S_2 \cdot f^{r_{\mathsf{ID}}} \; , \;\; \theta_3 = S_3 \cdot \gamma^{r_{\mathsf{ID}}} \cdot (\nu' \prod_{i=1}^{m} \nu_i^{\mu_i})^{r_0} \; , \;\; \theta_4 = S_4 \cdot f^{-r_0} \; \Big).$$

This is a valid unblinded signature in our scheme. The simulator can next simply apply the blinding procedure exactly as in the actual scheme, by multiplying the θ_i by powers of h and constructing the associated NIZK proof, and then give the resulting signature to the adversary.

Finally, the adversary gives the simulator a forgery σ on message M^*. The simulator first checks that the signature verifies, otherwise the adversary is not successful and the simulator can abort. Next, it sets out to trace the identity, ID^*, of the forgery. Let $\sigma = (\ldots, \sigma_2, \ldots)$. For each identity ID_i that was the object of an earlier query by the adversary—whether to request a private key for ID_i or a signature on (ID_i, M)—, the simulator recalls from the transcript of its interaction with the challenger the corresponding value $\tilde{K}_2$ or S_2, as the case may be. Let us denote by χ_{ID_i} the value in question, which in both cases equals $\tilde{g}^{s_{\mathsf{ID}_i}}$. For each such ID_i, the simulator tests whether,

$$(\sigma_2)^q \stackrel{?}{=} (\chi_{\mathsf{ID}_i})^q.$$

If the equality holds for some ID_i, we let $\mathsf{ID}^* = \mathsf{ID}_i$ be the traced identity. If either the key for ID^* or a signature on M^* by ID^* was previously requested by the adversary, the simulator can safely abort since the adversary produced a disqualifying signature. Otherwise, including the case where no identity ID^* could be determined, the adversary was successful and the simulator must proceed with its own forgery.

Since $T_1 = e(h, \pi_1)$, we know that the value T_1 defined in the *Verify* algorithm has order q in G_T. Let then $\delta \in \mathbb{Z}_n$ be an integer such that $\delta \equiv 0 \pmod{q}$ and $\delta \equiv 1 \pmod{p}$. It follows that,

$$e(\sigma_1, \sigma_2\, \Omega)^\delta = A^\delta = e(\tilde{g}, \tilde{g})^{\alpha\delta}\, e(f, f)^{\beta\delta} = e(\tilde{g}, \tilde{g})^\alpha = \tilde{A},$$

and hence, since it all happens in subgroups of order p at this point,

$$e(\sigma_1^\delta, \sigma_2^\delta\, \Omega) = \tilde{A}.$$

By an analogous reasoning, we also deduce that $T_2 \in G_{Tq}$, and thus,

$$e(\sigma_2, u)^\delta \cdot e(\sigma_3, g)^{-\delta} \cdot e(\sigma_4, v' \prod_{i=1}^{m} v_i^{\mu_i})^{-\delta} = e(\sigma_2, \tilde{u})^\delta \cdot e(\sigma_3, \tilde{g})^{-\delta} \cdot e(\sigma_4, \tilde{v}' \prod_{i=1}^{m} \tilde{v}_i^{\mu_i})^{-\delta} = 1,$$

which leaves us with a second equality,

$$e(\sigma_2^\delta, \tilde{u}) = e(\sigma_3^\delta, \tilde{g}) \cdot e(\sigma_4^\delta, \tilde{v}' \prod_{i=1}^{m} \tilde{v}_i^{\mu_i}).$$

We have just shown that $\sigma^* = (\sigma_1^\delta, \sigma_2^\delta, \sigma_3^\delta, \sigma_4^\delta)$ satisfied the verification conditions of the hybrid signature scheme in $\mathbb{Z}_p$, for message M^* and some unspecified (possibly unknown) identity ID^*. Thus, it suffices for the simulator to output σ^* as the forgery to win its own game with the challenger. This shows that our simulator will be successful whenever the adversary is.