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 [19], 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 [1–3, 5, 6, 9, 12–16, 25, 27, 29].

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, 16] or use bilinear groups [9, 11, 17]. 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 [13] 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 [23]. 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 [28]. 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 [11].

Groth [21] also gave a recent group signature scheme that was proven CCA-secure in the standard model under the decisional Linear assumption [9]. 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 [13] 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) [8] 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 [23], 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 [13] 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*)

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

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 [9] relax this definition by withholding access to the tracing oracle, thus mirroring the notion of IND-CPA security for encryption. We follow [9] 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 [26, 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, 27].

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 [10] 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 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 [24] 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 [10]. 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 [8] 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 [8]. 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 [8] 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 [8], 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 [8, §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 [30] 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 $\{s_1, \dots, s_{2^k}\} \subset \mathbb{Z}_p$ where $k = O(\log(\lambda))$. For convenience, we use sequential identifiers $\mathsf{ID} = 1, \dots, 2^k$ 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 $\boldsymbol{z} = (z_1, \dots, z_m) \in \mathbb{Z}_p^m$ are selected at random. The public parameters and master key are

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

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

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, $\{s_1, \dots, s_{2^k}\} \subset \mathbb{Z}_p$.

Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$**:** To create a private key for the identity s_{ID} associated with the user of index $1 \le \mathsf{ID} \le 2^k$, 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 $M = (\mu_1 \dots \mu_m) \in \{0,1\}^m$, 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

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

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 $M = (\mu_1 \dots \mu_m) \in \{0,1\}^m$, check whether

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

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 the full paper.

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 [12, 13], the other to Ateniese et al. [1].

The key difference with the earlier Boyen-Waters group signature scheme [12, 13], is that the earlier scheme relied on an all-purpose bit hiding technique due to Groth, Ostrovsky, and Sahai [23] 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,

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

Enroll$(\mathsf{PP}, \mathsf{MK}, \mathsf{ID})$**:** Suppose we wish to create a signing key for user ID, where $0 \le \mathsf{ID} < 2^k < p$. 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,

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

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),

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

Sign$(\mathsf{PP}, \mathsf{ID}, K_{\mathsf{ID}}, M)$**:** To sign a message $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$, 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,

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

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,

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

The final signature is output as:

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

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.

Next we state the security properties of our constant-size group signature scheme.

5.3 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 Γ_0 is the real group signature anonymity game, and Γ_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. The proofs of these two lemmas are given in the full paper.

5.4 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 is 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 the full paper.

6 CCA-Security

In the introduction we stated that the two primary drawbacks of the scheme of Boyen and Waters [12, 13] 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 (furthermore in a practical way), 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 [20]).

One tempting direction is to provably encrypt (in a simulation sound manner) the identity of the signer in a CCA-secure cryptosystems derived from one of the recent bilinear map-based IBE systems of Boneh and Boyen [7] combined with the techniques of Canetti, Halevi, and Katz [18]. Then we could allow the tracer to 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 the other 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 proof systems [22] derived from the decisional Linear assumption [9].

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. Efficient selective-ID secure identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–38. Springer-Verlag, 2004.
8. 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.
9. 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.
10. 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.
11. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of ACM CCS 2004*, pages 168–77. ACM Press, 2004.
12. Xavier Boyen and Brent Waters. Compact group signatures without random oracles. Cryptology ePrint Archive, Report 2005/381, 2005. `http://eprint.iacr.org/`.
13. 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.

14. Jan Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology—EUROCRYPT 1997*, Lecture Notes in Computer Science, pages 465–479. Springer-Verlag, 1997.
15. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *Proceedings of SCN 2004*, pages 120–133, 2004.
16. 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.
17. 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.
18. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*. Springer-Verlag, 2004.
19. 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.
20. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of STOC 1991*, pages 542–552, 1991.
21. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Proceedings of ASIACRYPT 2006*, Lecture Notes in Computer Science, pages 444–459. Springer-Verlag, 2006.
22. 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.
23. 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.
24. Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4), 2003.
25. 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.
26. 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/`.
27. 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.
28. Moni Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology—CRYPTO 2003*, Lecture Notes in Computer Science, pages 96–109. Springer-Verlag, 2003.
29. Dawn Xiaodong Song. Practical forward secure group signature schemes. In *ACM Conference on Computer and Communications Security—CCS 2001*, pages 225–234, 2001.
30. 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.