A Group Signature Scheme from Lattice Assumptions[*]

S. Dov Gordon[†] Jonathan Katz[†] Vinod Vaikuntanathan[‡]

Abstract

Group signature schemes allow users to sign messages on behalf of a group while (1) maintaining *anonymity* (within that group) with respect to an observer, yet (2) ensuring *traceability* of a signer (by the group manager) when needed. In this work we give the first construction of a group signature scheme based on lattices (more precisely, the *learning with errors* assumption), in the random oracle model. Toward our goal, we construct a new algorithm for sampling a random superlattice of a given modular lattice *together with a short basis*, that may be of independent interest.

1 Introduction

Group signature schemes [16] allow users to sign messages on behalf of a group administered by some manager. The group is initialized by having the group manager generate master public and secret keys; upon admission to the group, a user is given a personal secret key that is derived from the master secret key by the manager. A member of the group can sign a message using their personal secret key, enabling anyone who knows the master public key to verify that *some* group member signed the message. Roughly, group signatures are required to satisfy two seemingly contradictory requirements: given some legitimate group signature σ, the group manager should be able to determine which member of the group issued σ (*traceability*), but no one other than the group manager should be able to determine any information about the signer (*anonymity*). Group signatures have proven to be a popular primitive, and since their introduction several constructions have been proposed both with random oracles [5, 6, 13, 10, 14, 22] and without [7, 9, 4, 11, 12, 21].

While there exist constructions of group signature schemes based on trapdoor permutations [7, 9], such schemes serve only as proofs of feasibility and are far from practical. On the other hand, practical schemes are based on a relatively small set of assumptions: namely, the strong RSA assumption [5, 6, 13, 22] and various assumptions related to groups having an associated bilinear map [10, 14, 4, 11, 12, 21]. In this work we show the first construction of a group signature scheme from assumptions related to *lattices*. The use of lattice-based assumptions in cryptography has seen a flurry of activity in recent years. In part, this is due to a general desire to expand the set of assumptions on which cryptosystems can be based (i.e., beyond the standard set of assumptions related to the hardness of factoring and solving the discrete logarithm problem). Relying on lattice-based assumptions offers several concrete advantages as well: such assumptions

[*] Work done while the authors were at IBM Research.

[†] Department of Computer Science, University of Maryland. Research supported by NSF grants #0627306 and #0716651. Email: {gordon,jkatz}@cs.umd.edu

[‡] Microsoft Research. Email: vinod@microsoft.com

are appealing because of the known worst-case/average-case connections between lattice problems, and also because lattice problems are currently immune to quantum attacks. Even restricting to classical attacks, the best-known algorithms for solving several lattice problems require exponential time (in contrast to the sub-exponential algorithms known, e.g., for factoring). Finally, relying on lattices can potentially yield efficient constructions because the basic lattice operations manipulate relatively small numbers and are inherently parallelizable.

While our resulting construction is less efficient than existing schemes based on number-theoretic assumptions, our construction is significantly more efficient than the generic approaches of [7, 9] that rely on NIZK proofs based on a Karp reduction to some NP-complete language. We remark that although Peikert and Vaikuntanathan [27] construct efficient NIZK proofs for specific lattice problems, their results are not directly applicable to the existing generic constructions.

1.1 Our Techniques

Our construction combines ideas from several different works, tying these together using a new technical tool described below. At a high level, our group signature scheme follows a template similar (but not identical) to that of Bellare et al. [7]. The master public key in our scheme includes a public key pk_E for a public-key encryption scheme, along with n signature verification keys $pk_1, \ldots, pk_N$. The personal secret key given to the ith group member is sk_i, the signing key corresponding to pk_i. To sign a message M, the group member (1) signs M using sk_i; (2) encrypts the resulting signature using pk_E; and then (3) provides a NIZK proof of well-formedness (namely, that the given ciphertext encrypts a signature on M relative to one of the pk_i). This implies anonymity (since no one other than the group manager knows the decryption key sk_E corresponding to pk_E), yet ensures traceability because the group manager can decrypt the ciphertext that is included as part of any valid group signature.

To instantiate this approach using lattice-based assumptions, we need to identify candidate signature and encryption schemes along with an appropriate NIZK proof system. While constructions of the former primitives based on lattices are known, we do not currently have constructions of NIZK for all of NP from lattice-based assumptions and we therefore have to tailor our scheme so that it can rely on (efficient) NIZK proofs for some *specific* language. This is explained in more detail in what follows.

For the underlying signature scheme we use the GPV signature scheme [19] that works roughly as follows. The public key is a basis $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for a random lattice. To sign a message M, the signer uses a trapdoor $\mathbf{T}$ to find a "short" vector $\mathbf{e} \in \mathbb{Z}^m$ with $\mathbf{A}\mathbf{e} = H(M)$ (where H is a hash function modeled as a random oracle). Under suitable assumptions, finding such a short vector $\mathbf{e}$ without the trapdoor is hard.

We encrypt the resulting signature using what can be viewed as a non-standard variant of the Regev encryption scheme [28]. Given a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, viewed as a public key, we encrypt $\mathbf{e} \in \mathbb{Z}^m$ by choosing a random vector $\mathbf{s} \in \mathbb{Z}_q^n$ and outputting the ciphertext $\mathbf{z} = \mathbf{B}^T\mathbf{s} + \mathbf{e}$. Effectively, $\mathbf{e}$ here is being used as the noise in an instance of the "learning with errors" (LWE) problem [28]. Before going further, we stress that this "encryption scheme" is *not* semantically secure. However, it turns out that we need something much weaker than semantic security in order to prove anonymity of our scheme; roughly, all we need is that the encryption of a uniformly random $\mathbf{e} \in \mathbb{Z}_q^m$ is computationally indistinguishable from the encryption of a vector $\mathbf{e}$ chosen from a certain discrete Gaussian distribution. We defer further discussion to Section 3.

As described thus far, our group signature scheme would have a master public key consisting of verification keys $\mathbf{A}_1, \ldots, \mathbf{A}_N$ along with an encryption key $\mathbf{B}$; a signature would include $\mathbf{z} = \mathbf{B}^T\mathbf{s} + \mathbf{e}$, where $\mathbf{e}$ is such that $\mathbf{A}_i\mathbf{e} = H(M)$ for some i, along with a proof of well-formedness of the ciphertext $\mathbf{z}$. Constructing the proof of well-formedness turns out to be the most difficult aspect of our work, and we must modify our scheme a bit in order to make such a proof (reasonably) efficient. (In doing so, we also rely on specific properties of the GPV signature scheme.) We change our scheme as follows: Now, the master public key contains N verification keys $\mathbf{A}_1, \ldots, \mathbf{A}_N$ (as before) and also N encryption keys $\mathbf{B}_1, \ldots, \mathbf{B}_N$. To sign a message M, user i computes a real signature $\mathbf{e}_i$ (using the trapdoor associated with $\mathbf{A}_i$) and "pseudo-signatures" $\mathbf{e}_j$ for all $j \neq i$. Each "pseudo-signature" $\mathbf{e}_j$ has the property that $\mathbf{A}_j\,\mathbf{e}_j = H(M)$, however $\mathbf{e}_j$ *is not short* (and thus not a valid signature). All the $\{\mathbf{e}_j\}_{j=1}^N$ are then encrypted as before, with each $\mathbf{e}_j$ being encrypted using $\mathbf{B}_j$ to give a ciphertext $\mathbf{z}_j$. We then have the signer provide a proof that (1) each $\mathbf{z}_j$ encrypts a correct pseudo-signature with respect to $\mathbf{A}_j$, and (2) at least one of these pseudo-signatures is *short* (and hence, in fact, a valid signature). Further details are given next.

To provide a way for the signer to prove that every ciphertext $\mathbf{z}_j$ encrypts a pseudo-signature, we develop a new technical tool that we believe to be of independent interest: a way to sample a basis for an *orthogonal lattice* with its associated trapdoor.[1] Specifically, we show a technique that, given a matrix $\mathbf{B}$, generates $(\mathbf{A}, \mathbf{T})$ such that $\mathbf{A}\mathbf{B}^T = 0 \pmod q$ and $\mathbf{T}$ is still a "good trapdoor" (in the sense required for GPV signatures) for $\mathbf{A}$. If we use matrices $\{\mathbf{A}_i\}$ generated in this way as verification keys in the group signature scheme described earlier, then it is possible to verify that a given ciphertext $\mathbf{z}_j$ encrypts a pseudo-signature with respect to $\mathbf{A}_j$ by checking whether $\mathbf{A}_j \cdot \mathbf{z}_j \overset{?}{=} H(M)$. This works because

$$\mathbf{A}_j \cdot \mathbf{z}_j = \mathbf{A}_j \cdot \left(\mathbf{B}_j^T\mathbf{s}_j + \mathbf{e}_j\right) = \mathbf{A}_j \cdot \mathbf{e}_j = H(M)$$

by construction.

The only thing that remains is to provide a proof that at least one of the $\mathbf{z}_j$ encrypts a vector $\mathbf{e}_j$ that is also *short*. This translates to proving that at least one of the vectors $\mathbf{z}_j = \mathbf{B}_j^T\mathbf{s}_j + \mathbf{e}_j$ is "close to" the lattice generated by the columns of $\mathbf{B}_j^T$. This can be done using the (statistical) zero-knowledge protocol of Micciancio and Vadhan [24], coupled with standard techniques [17, 18] for making the proof witness indistinguishable and noninteractive in the random oracle model.

1.2 Outline of the Paper

We introduce some notation and review the necessary background on lattices in Section 2. For the reader who is already familiar with lattices, we highlight the following aspects of our treatment that are new to this work:

- In Section 2.2 (cf. Lemma 2) and in the rest of the paper, we consider the LWE problem under a non-standard error distribution. Peikert [26] shows that the hardness of the LWE problem under this distribution is implied by standard hardness results.

- In Section 2.4 we describe a technique for sampling a basis for an *orthogonal* lattice and its associated trapdoor.

We turn to group signatures in Section 3. We review the standard definitions of security for group signature schemes in Section 3.1, describe our construction in Section 3.2, and prove anonymity and traceability in Sections 3.3 and 3.4.

[1] For our definition of an orthogonal lattice, see Section 2.

2 Preliminaries on Lattices

Throughout, we use n for the security parameter; other parameters are taken to be functions of n. When we say "statistically close" we mean "within statistical difference negligible in n."

We review some basic properties of lattices as used in prior work. This section is included mainly to fix notation and ideas, and we refer to the original papers (cited below) for further exposition.

We use bold lower-case letters (e.g., $\mathbf{x}$) to denote vectors, and bold upper-case letters (e.g., $\mathbf{B}$) to denote matrices. (Our vectors are always column vectors.) We let $||\mathbf{x}||$ denote the Euclidean (i.e., ℓ_2) norm of the vector $\mathbf{x}$, and let $||\mathbf{B}||$ denote the maximum of the Euclidean norms of the columns of $\mathbf{B}$; i.e., if $\mathbf{B} = (\mathbf{b}_1|\cdots|\mathbf{b}_n)$ then $||\mathbf{B}|| \stackrel{\text{def}}{=} \max_i ||\mathbf{b}_i||$. If $x \in \mathbb{R}$, then $\lfloor x \rceil$ denotes the rounding of x to the nearest integer.

For q an integer, $\mathbb{Z}_q$ denotes the standard group of integers modulo q. We extend modular arithmetic to the reals in the obvious way: for example, for $q \in \mathbb{Z}^+$ and $x \in \mathbb{R}$ we use $x \bmod q$ to represent the unique real number $y \in [0, q)$ such that $x - y$ is an integer multiple of q. Finally, we define a notion of distance between elements in $\mathbb{Z}_q$ in the natural way: given $x, y \in \mathbb{Z}_q$, their distance $|x - y|$ is defined by mapping $(x - y) \bmod q$ to the set of integers $\{-\lfloor q/2 \rfloor, \ldots, \lceil q/2 \rceil - 1\}$ and then taking the absolute value of the result. We define the distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{Z}_q^m$ in an analogous way, namely, $||\mathbf{x} - \mathbf{y}||^2 = \sum_{i=1}^m |\mathbf{x}_i - \mathbf{y}_i|^2$.

Fixing q and given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define the m-dimensional lattice $\Lambda(\mathbf{A}^T)$ as

$$\Lambda(\mathbf{A}^T) \stackrel{\text{def}}{=} \left\{ \mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} \equiv \mathbf{A}^T \mathbf{s} \pmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^n \right\}.$$

In other words, $\Lambda(\mathbf{A}^T)$ consists of all linear combinations of the columns of $\mathbf{A}^T$, shifted by integer multiples of the modulus q. We define[2] the *orthogonal lattice* $\Lambda^\perp(\mathbf{A})$ as

$$\Lambda^\perp(\mathbf{A}) \stackrel{\text{def}}{=} \{ \mathbf{w} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{w} = \mathbf{0} \pmod q \}$$

An observation worth noting is that for any $\mathbf{y} \in \Lambda(\mathbf{A}^T)$ and $\mathbf{w} \in \Lambda^\perp(\mathbf{A})$, it holds that $\mathbf{y}^T \mathbf{w} = 0 \bmod q$. Finally, for a vector $\mathbf{z} \in \mathbb{Z}_q^m$ we define

$$\mathsf{dist}(\Lambda(\mathbf{A}^T), \mathbf{z}) \stackrel{\text{def}}{=} \min_{\mathbf{s} \in \mathbb{Z}_q^n} ||(\mathbf{A}^T \mathbf{s} - \mathbf{z}) \bmod q||.$$

2.1 Gaussian Error Distributions

The one-dimensional (continuous) Gaussian distribution over $\mathbb{R}$, parameterized by $s \in \mathbb{R}^+$, is defined by the density function

$$\forall x \in \mathbb{R}: \qquad D_s(x) = 1/s \cdot \exp(-\pi (x/s)^2).$$

The m-dimensional continuous Gaussian distribution is defined in a similar way, by the density function $D_s(\mathbf{x}) = 1/s^m \cdot \exp(-\pi(||\mathbf{x}||/s)^2)$. Finally, we denote by $D_{s,\mathbf{c}}$ the m-dimensional continuous Gaussian distribution centered at the point $\mathbf{c} \in \mathbb{R}^m$. i.e., $D_{s,\mathbf{c}}(\mathbf{x}) = 1/s^m \cdot \exp(-\pi(||\mathbf{x} - \mathbf{c}||/s)^2)$. In this work we always let D_s (resp., $D_{s,\mathbf{c}}$) denote *truncated* Gaussian distributions, i.e., the Gaussian distribution conditioned on $|x| < s \cdot \omega(\sqrt{\log n})$ (resp., $||\mathbf{x} - \mathbf{c}|| < s \cdot \omega(\sqrt{\log n})$). The truncated and non-truncated distributions are statistically close, and we drop the word "truncated" from now on.

[2] Our definition of an orthogonal lattice differs from that given in some previous work. $\Lambda^\perp(\mathbf{A})$, as we define it, is merely a scaling of $(\Lambda(\mathbf{A}^T))^*$, the dual of $\Lambda(\mathbf{A}^T)$. In particular, $\Lambda^\perp(\mathbf{A}) = q \cdot (\Lambda(\mathbf{A}^T))^*$.

Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. The *discrete Gaussian distribution* $D_{\Lambda,s,\mathbf{c}}$ is the m-dimensional Gaussian distribution centered at $\mathbf{c}$, but with support restricted to the lattice Λ. (We write $D_{\Lambda,s}$ as shorthand for $D_{\Lambda,s,\mathbf{0}}$.) Formally, the density function of the discrete Gaussian distribution is

$$\forall \mathbf{x} \in \Lambda : \qquad D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{D_{s,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{y}\in\Lambda} D_{s,\mathbf{c}}(\mathbf{y})} \, .$$

Gentry et al. [19] show that given a basis $\mathbf{B}$ for Λ, this distribution can be sampled efficiently (to within negligible statistical distance) for $s \geq ||\mathbf{B}|| \cdot \omega(\sqrt{\log n})$.

2.2 The Learning with Errors Problem

The "learning with errors" (LWE) problem was introduced by Regev [28] as a generalization of the "learning parity with noise" problem. We describe the problem in a form suitable for our applications in this paper.

Fix a positive integer n, integers $m \geq n$ and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{R}^m$. (For our purposes, χ will always be a product distribution in each coordinate.) Define the following two distributions over $\mathbb{Z}_q^{n\times m} \times [0,q)^m$:

- $\mathsf{LWE}_{m,q,\chi}(\mathbf{s})$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, sampling $\mathbf{e} \leftarrow \chi$, and outputting $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q)$.
- $U_{m,q}$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and uniform $\mathbf{y} \in [0,q)^m$, and outputting $(\mathbf{A}, \mathbf{y})$.

The decisional variant of the LWE problem (relative to the distribution χ) can be stated informally as the problem of distinguishing between $\mathsf{LWE}_{m,q,\chi}(\mathbf{s})$ (for a uniform, unknown $\mathbf{s}$) and $U_{m,q}$. Formally, for m, q, and χ that may depend on n we say the $\mathsf{LWE}_{m,q,\chi}$ *problem is hard* if the following is negligible for any probabilistic polynomial-time algorithm D:

$$\left|\Pr[\mathbf{s} \leftarrow \mathbb{Z}_q^n; (\mathbf{A},\mathbf{y}) \leftarrow \mathsf{LWE}_{m,q,\chi}(\mathbf{s}) : D(\mathbf{A},\mathbf{y}) = 1] - \Pr[(\mathbf{A},\mathbf{y}) \leftarrow U_{m,q} : D(\mathbf{A},\mathbf{y}) = 1]\right| .$$

A standard setting for the LWE problem considers the error distribution $D_{\alpha q}$. We write $\mathsf{LWE}_{m,q,\alpha}(\mathbf{s})$ as an abbreviation for $\mathsf{LWE}_{m,q,D_{\alpha q}}(\mathbf{s})$. Evidence for the hardness of the $\mathsf{LWE}_{m,q,\alpha}$ problem comes from a result of Regev [28], who gave a *quantum* reduction from approximating certain lattice problems in the worst case to solving $\mathsf{LWE}_{m,q,\alpha}$, subject to the condition that $\alpha{\cdot}q > 2\sqrt{n}$. Peikert [25] later gave a *classical* reduction with similar parameters. For our purposes, we note that the $\mathsf{LWE}_{m,q,\alpha}$ problem is believed to be hard (given the state-of-the-art in lattice algorithms) for any $m, q = \mathsf{poly}(n)$ and $\alpha = 1/\mathsf{poly}(n)$ subject to $\alpha \cdot q > 2\sqrt{n}$.

A second error distribution that can be considered for the LWE problem[3] — and the one that we will use in this paper — is the discrete Gaussian distribution $D_{\mathbb{Z}^m,\alpha q}$. (We write $\widehat{\mathsf{LWE}}_{m,q,\alpha}$ as an abbreviation for $\mathsf{LWE}_{m,q,D_{\mathbb{Z}^m,\alpha q}}$.) Although this distribution may seem similar to a discretized (rounded) version of $D_{\alpha'}$ (for appropriate choice of α'), these distributions are statistically *far* from each other and thus we cannot immediately conclude anything about the hardness of the LWE problem with respect to one distribution from hardness of the LWE problem with respect to

[3] When using a discrete error distribution χ over $\mathbb{Z}^m$ (rather than a continuous distribution over $\mathbb{R}^m$), the LWE problem is to distinguish $\mathsf{LWE}_{m,q,\chi}$ from the uniform distribution over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ (rather than $\mathbb{Z}_q^{n\times m} \times [0,q)^m$).

the other. Fortunately, a recent result of Peikert [26] can be used to show that hardness of the $\widehat{\mathsf{LWE}}_{m,q,\alpha\sqrt{2}}$ problem is implied by hardness of the $\mathsf{LWE}_{m,q,\alpha}$ problem. The following is a special case of Peikert's result [26, Theorem 1] (for the standard definition of $\eta_\epsilon(\Lambda)$, the *smoothing parameter* of the lattice Λ, see [26]):

Lemma 1 *Let Λ be a lattice, and let $s, s_1, s_2 > 0$ and $\epsilon \in (0, 1/2]$ be such that $s^2 = s_1^2 + s_2^2$ and $s_1 \geq \eta_\epsilon(\Lambda)$. Consider the experiment in which we first sample $\mathbf{e} \leftarrow D_{s_2}$ and then sample $\mathbf{e}' \leftarrow \mathbf{e} + D_{\Lambda - \mathbf{e}, s_1}$. Then the distribution of $\mathbf{e}'$ is within statistical distance 8ϵ of $D_{\Lambda,s}$.*

Below, we take $\Lambda = \mathbb{Z}^m$ for which $\eta_\epsilon(\mathbb{Z}^m) = O(\sqrt{\log \epsilon^{-1}})$ (see [23, Lemma 3.3]). When $s_1 = \Omega(\sqrt{\log n})$, as will be the case for our applications, we can then take ϵ to be negligible and thus the two distributions considered in the lemma are statistically close. With this in place we can now prove our desired result:

Lemma 2 *For any $m = m(n), q = q(n), \alpha = \alpha(n)$ satisfying $\alpha q = \omega(\sqrt{\log n})$, hardness of the $\mathsf{LWE}_{m,q,\alpha}$ problem implies hardness of the $\widehat{\mathsf{LWE}}_{m,q,\alpha\sqrt{2}}$ problem.*

Proof We show an efficient transformation T that takes as input $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times m} \times [0, q)^m$ and has the following properties:

- If $(\mathbf{A}, \mathbf{y})$ is uniform over $\mathbb{Z}_q^{n \times m} \times [0, q)^m$ then the output $T(\mathbf{A}, \mathbf{y})$ is uniform over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.
- If $(\mathbf{A}, \mathbf{y})$ is distributed according to $\mathsf{LWE}_{m,q,\alpha}(\mathbf{s})$ then the distribution $T(\mathbf{A}, \mathbf{y})$ is statistically close to $\widehat{\mathsf{LWE}}_{m,q,\alpha\sqrt{2}}(\mathbf{s})$.

The lemma follows immediately from these two properties.

The transformation T works as follows. Given $(\mathbf{A}, \mathbf{y})$, it samples a vector $\mathbf{w} \leftarrow D_{\mathbb{Z}^m - \mathbf{y}, \alpha q}$ and outputs $(\mathbf{A}, \mathbf{y} + \mathbf{w} \bmod q)$.

Say $(\mathbf{A}, \mathbf{y})$ is distributed uniformly over $\mathbb{Z}_q^{n \times m} \times [0, q)^m$. Note that $\mathbf{y} + \mathbf{w}$ is always an integer vector, and the distribution $D_{\mathbb{Z}^m - \mathbf{y}, \alpha q}$ depends only on the fractional part of each entry of $\mathbf{y}$. In other words, the integer part of each entry in $\mathbf{y}$ ensures that each entry of $\mathbf{y} + \mathbf{w} \bmod q$ is uniform in $\mathbb{Z}_q$. It follows that $(\mathbf{A}, \mathbf{y} + \mathbf{w} \bmod q)$ is distributed uniformly over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.

On the other hand, say $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q$ where $\mathbf{e} \leftarrow D_{\alpha q}$. Since we have $\mathbf{A}^T\mathbf{s} \in \mathbb{Z}^m$, sampling $\mathbf{w} \leftarrow D_{\mathbb{Z}^m - \mathbf{y}, \alpha q}$ is equivalent to sampling $\mathbf{w} \leftarrow D_{\mathbb{Z}^m - \mathbf{e}, \alpha q}$. Using Lemma 1, sampling $\mathbf{e} \leftarrow D_{\alpha q}$ and then setting $\mathbf{e}' = \mathbf{e} + \mathbf{w}$ for $\mathbf{w} \leftarrow D_{\mathbb{Z}^m - \mathbf{e}, \alpha q}$ yields a distribution for $\mathbf{e}'$ that is statistically close to sampling $\mathbf{e}' \leftarrow D_{\mathbb{Z}^m, \alpha q\sqrt{2}}$. We conclude that the output $T(\mathbf{A}, \mathbf{y}) = \big(\mathbf{A}, \mathbf{A}^T\mathbf{s} + (\mathbf{e} + \mathbf{w}) \bmod q\big)$ has distribution statistically close to that of $\widehat{\mathsf{LWE}}_{m,q,\alpha\sqrt{2}}(\mathbf{s})$. ■

2.3 Trapdoor Functions and the GPV Signature Scheme

Ajtai [2] and Alwen and Peikert [3] show algorithms that generate an almost uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a "trapdoor" matrix $\mathbf{T} \in \mathbb{Z}^{m \times m}$ satisfying the following conditions:

Lemma 3 ([3]) *There is a* PPT *algorithm* $\mathsf{TrapSamp}$ *that, on input 1^n, 1^m, q with $q \geq 2$ and $m \geq 8n \log q$, outputs matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} \in \mathbb{Z}^{m \times m}$ such that the distribution on $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ and, with probability all but negligible in n:*

- *the columns of* $\mathbf{T}$ *form a basis of the lattice* $\Lambda^{\perp}(\mathbf{A})$*, implying in particular* $\mathbf{A}\cdot\mathbf{T} = 0 \pmod{q}$*,*
- $\|\mathbf{T}\| = O(n \log q)$ *and* $\|\widetilde{\mathbf{T}}\| = O(\sqrt{n \log q})$*. (Here,* $\widetilde{\mathbf{T}}$ *is the Gram-Schmidt orthogonalization of* $\mathbf{T}$*.)*

Given an "LWE instance" $(\mathbf{A}, \mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q)$ for a "short" vector $\mathbf{e}$, knowledge of $\mathbf{T}$ can be used to recover $\mathbf{s}$. Specifically, if $||\mathbf{T}|| < L$ and $\mathbf{e}$ is drawn from $D_{\alpha q}$ for $\alpha \leq 1/(L \cdot \omega(\sqrt{\log n}))$, then $\mathbf{s}$ can be easily recovered. This is done by first computing

$$\begin{aligned} \mathbf{T}^T\mathbf{y} \bmod q = \mathbf{T}^T(\mathbf{A}^T\mathbf{s} + \mathbf{e}) \bmod q &= (\mathbf{A}\mathbf{T})^T\mathbf{s} + \mathbf{T}^T\mathbf{e} \bmod q \\ &= \mathbf{T}^T\mathbf{e} \bmod q. \end{aligned}$$

Since $\mathbf{T}$ and $\mathbf{e}$ contain only "small" entries, each entry of the vector $\mathbf{T}^T\mathbf{e}$ is smaller than q and thus $\mathbf{T}^T\mathbf{e} \bmod q$ is equal to $\mathbf{T}^T\mathbf{e}$ (over the integers). Multiplying by $(\mathbf{T}^T)^{-1}$ thus gives $\mathbf{e}$, after which it is easy to recover $\mathbf{s}$.

Gentry, Peikert, and Vaikuntanathan [19] show how to use the trapdoor sampling procedure described above to construct a one-way preimage-sampleable function. This can then be turned into a digital signature scheme using an "FDH-like" construction [8]. (See [19] for a formal definition of preimage-sampleable functions and the construction of the signature scheme.) Here, we describe how the preimage-sampleable function works.

Take $q = \mathsf{poly}(n)$, $m \geq 8n \log q$, and $s = \omega(\sqrt{n \log q \log n})$. The one-way preimage-sampleable function is defined by the following algorithms:

- $\mathsf{GPVGen}(1^n)$ runs $\mathsf{TrapSamp}(1^n, 1^m, q)$ to obtain $(\mathbf{A}, \mathbf{T})$. The matrix $\mathbf{A}$ (and q) defines the function $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$, with domain $\{\mathbf{e} \in \mathbb{Z}^m : ||\mathbf{e}|| \leq s\sqrt{m}\}$ and range $\mathbb{Z}_q^n$. Hardness of inversion is with respect to the distribution $D_{\mathbb{Z}^m, s}$ over the domain.
- The trapdoor inversion algorithm $\mathsf{GPVInvert}(\mathbf{A}, \mathbf{T}, s, \mathbf{u})$ samples from $f_{\mathbf{A}}^{-1}(\mathbf{u})$ as follows: first, it computes (using standard linear algebra) $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{t} = \mathbf{u} \pmod{q}$. (Except for a negligible fraction of $\mathbf{A}$, such a $\mathbf{t}$ always exists.) Then it samples and outputs $\mathbf{e} \leftarrow D_{\Lambda^{\perp}(\mathbf{A}) + \mathbf{t}, s}$.

The above is one-way if GapSVP is worst-case hard for some suitable approximation factor [1].

2.4 Sampling a Random Superlattice with a Short Basis

We show a variant of the trapdoor sampling algorithm described in Lemma 3. In our variant, the algorithm is additionally given a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ whose columns span $\mathbb{Z}_q^n$, and should output a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (with an associated trapdoor $\mathbf{T} \in \mathbb{Z}^{m \times m}$) satisfying the additional requirement that $\Lambda(\mathbf{B}^T) \subseteq \Lambda^{\perp}(\mathbf{A})$. Thus, $\mathbf{A}$ is sampled under the condition that $\Lambda^{\perp}(\mathbf{A})$ is a superlattice of $\Lambda(\mathbf{B}^T)$ or, equivalently, $\mathbf{A}\mathbf{B}^T = \mathbf{0} \pmod{q}$. We also require that the trapdoor $\mathbf{T}$ be a "random basis" for $\Lambda^{\perp}(\mathbf{A})$, in the sense that the columns of $\mathbf{T}$ are distributed according to a discrete Gaussian distribution over $\Lambda^{\perp}(\mathbf{A})$.

Overview of the construction. Say we are given a matrix $\mathbf{B}$ whose columns span $\mathbb{Z}_q^n$. Write

$$\mathbf{B}^T = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix},$$

with $\mathbf{B}_2$ a square, invertible matrix of dimension $n \times n$. (By the stated assumption on $\mathbf{B}$, such a decomposition can always be found, permuting the rows of $\mathbf{B}^T$ if necessary.) We generate a matrix $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$ in two steps. We generate the first component $\mathbf{A}_1$ using the $\mathsf{TrapSamp}$ protocol. Recall, this returns a matrix that is statistically close to uniform, along with an associated trapdoor $\mathbf{T}_1$. With $\mathbf{A}_1$ fixed, the second component $\mathbf{A}_2$ is constrained to a fixed value by the requirement that $\mathbf{A}\mathbf{B}^T = \mathbf{0} \pmod q$; we generate $\mathbf{A}_2$ by solving the linear equations that define this constraint.

We then need to extend $\mathbf{T}_1$ into a trapdoor $\mathbf{T}$ whose columns are "short" and such that $\mathbf{A}\cdot\mathbf{T} = \mathbf{0}$. Here we rely on a recent techniques of Cash et al. [15], which allows us to extend $\mathbf{T}_1$ into a basis $\mathbf{T}$ for $\Lambda^\perp(\mathbf{A})$. We use a second technique from their work to "randomize" $\mathbf{T}$ before outputting it.

Lemma 4 *There is a* PPT *algorithm* $\mathsf{SuperSamp}$ *that on input* $1^n, 1^m, q$ *(with* $q \geq 2$ *and* $m \geq n + 8n \log q$*), and* $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ *whose columns span* $\mathbb{Z}_q^n$*, outputs* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and* $\mathbf{T} \in \mathbb{Z}^{m\times m}$ *such that* $\mathbf{A}\mathbf{B}^T = \mathbf{0} \pmod q$ *and the distribution on* $\mathbf{A}$ *is statistically close to uniform over* $\mathbb{Z}_q^{n\times m}$ *subject to this condition. Moreover, with probability all but negligible in* n*:*

- *the columns of* $\mathbf{T}$ *form a basis of the lattice* $\Lambda^\perp(\mathbf{A})$*, implying in particular* $\mathbf{A}\cdot\mathbf{T} = 0 \pmod q$*,*
- $||\widetilde{\mathbf{T}}|| = O(\log n \cdot \sqrt{mn\log q})$*.*

Proof Let $m_2 = n$ and $m_1 = m - m_2$. Write

$$\mathbf{B}^T = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix},$$

where $\mathbf{B}_1 \in \mathbb{Z}_q^{m_1\times n}$ and $\mathbf{B}_2 \in \mathbb{Z}_q^{m_2\times n}$, and furthermore the (square) matrix $\mathbf{B}_2$ has full rank over $\mathbb{Z}_q$. (By the stated assumption on $\mathbf{B}$, such a decomposition can always be found, permuting the rows of $\mathbf{B}^T$ if necessary.)

Algorithm $\mathsf{SuperSamp}$ works as follows:

1. Compute $(\mathbf{A}_1, \mathbf{T}_1) \leftarrow \mathsf{TrapSamp}(1^n, 1^{m_1}, q)$. If the columns of $\mathbf{A}_1$ do not span $\mathbb{Z}_q^n$, output $\perp$. (This occurs only with negligible probability.) Otherwise, let $\mathbf{A}_2 \in \mathbb{Z}_q^{n\times m_2}$ be the unique matrix satisfying
$$\mathbf{A}_2\mathbf{B}_2 = -\mathbf{A}_1\mathbf{B}_1 \pmod q.$$
Since $\mathbf{B}_2$ is invertible, $\mathbf{A}_2$ can be computed as $-\mathbf{A}_1\mathbf{B}_1\mathbf{B}_2^{-1} \bmod q$. Let $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$.

2. Extend $\mathbf{T}_1 \in \mathbb{Z}_q^{m_1\times m_1}$ into basis $\mathbf{T}' \in \mathbb{Z}_q^{m\times m}$ for $\Lambda^\perp(\mathbf{A})$ using the $\mathsf{ExtBasis}$ algorithm of Cash et al. [15].

3. Randomize $\mathbf{T}'$ into a "random basis" $\mathbf{T}$ by applying the $\mathsf{RandBasis}$ algorithm of Cash et al. [15] to $\mathbf{T}'$, using $s = ||\widetilde{\mathbf{T}'}|| \cdot \log n$. Output $\mathbf{A}$ and $\mathbf{T}$.

We now verify that this algorithm satisfies the required properties. First observe that

$$\mathbf{A}\mathbf{B}^T = \mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_2 = \mathbf{A}_1\mathbf{B}_1 - \mathbf{A}_1\mathbf{B}_1 = \mathbf{0} \pmod q.$$

The claim regarding the distribution of $\mathbf{A}$ follows directly from the construction and the fact that $\mathbf{A}_1$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m_1}$. Properties of $\mathsf{TrapSamp}$ guarantee that (except with negligible probability) $\mathbf{T}_1$ is a basis for $\Lambda^\perp(\mathbf{A}_1)$; results of Cash et al. [15, Lemma 3] then imply that $\mathbf{T}'$ is a basis for $\Lambda^\perp(\mathbf{A})$, and so $\mathbf{T}$ is as well [15, Lemma 4].

Finally, we have

$$\begin{aligned}
||\widetilde{\mathbf{T}}|| &\leq ||\widetilde{\mathbf{T}'}|| \cdot \sqrt{m}\,\log n \ \ \text{(by properties of } \mathsf{RandBasis}\text{)} \\
&= ||\widetilde{\mathbf{T}}_1|| \cdot \sqrt{m}\,\log n \ \ \text{(by properties of } \mathsf{ExtBasis}\text{)} \\
&= O(\sqrt{n \log q} \cdot \sqrt{m}\,\log n) \ \ \text{(by properties of } \mathsf{TrapSamp}\text{)},
\end{aligned}$$

where the final equality holds with all but negligible probability. The lemma follows. ■

We will also use the following result regarding our algorithm $\mathsf{SuperSamp}$:

Lemma 5 *The distributions*

$$\left\{ \begin{array}{c} \mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}; \\ (\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}) \end{array} : (\mathbf{A}, \mathbf{T}, \mathbf{B}) \right\}$$

and

$$\left\{ \begin{array}{c} (\mathbf{A}, \mathbf{T}') \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q); \\ \mathbf{T} \leftarrow \mathsf{RandBasis}(\mathbf{T}'); \\ (\mathbf{B}, \mathbf{S}) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}) \end{array} : (\mathbf{A}, \mathbf{T}, \mathbf{B}) \right\}$$

are statistically close.

Proof In the first distribution, $\mathbf{B}$ is uniform and $\mathbf{A}$ is statistically close to uniform subject to the constraint $\mathbf{A}\mathbf{B}^T = \mathbf{0} \pmod q$; in the second distribution, $\mathbf{A}$ is statistically close to uniform and $\mathbf{B}$ is statistically close to uniform subject to the constraint $\mathbf{B}\mathbf{A}^T = \mathbf{0} \pmod q$. Thus, the marginal distributions of $(\mathbf{A}, \mathbf{B})$ are statistically close.

The lemma follows because, in both distributions, $\mathbf{T}$ is the result of applying the $\mathsf{RandBasis}$ algorithm to a basis for $\Lambda^\perp(\mathbf{A})$. ■

2.5 Efficient NIWI Proofs for Lattice Problems

Let $\mathbf{B}_1, \ldots, \mathbf{B}_N \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{z}_1, \ldots, \mathbf{z}_N \in \mathbb{Z}_q^m$, and fix some $\gamma = \gamma(n)$. In this section we briefly describe how it is possible to construct a noninteractive witness-indistinguishable (NIWI) proof (in the random oracle model) for the gap language $L_{s,\gamma} = (L_{YES}, L_{NO})$ defined by:

$$\begin{aligned}
L_{YES} &= \left\{ \left(\begin{array}{c} \mathbf{B}_1, \ldots, \mathbf{B}_N \\ \mathbf{z}_1, \ldots, \mathbf{z}_N \end{array} \right) \middle| \ \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [N] \ : \ ||\mathbf{z}_i - \mathbf{B}_i^T \mathbf{s}|| \leq s\sqrt{m} \right\} \\
L_{NO} &= \left\{ \left(\begin{array}{c} \mathbf{B}_1, \ldots, \mathbf{B}_N \\ \mathbf{z}_1, \ldots, \mathbf{z}_N \end{array} \right) \middle| \ \forall \mathbf{s} \in \mathbb{Z}_q^n \text{ and } i \in [N] \ : \ ||\mathbf{z}_i - \mathbf{B}_i^T \mathbf{s}|| > \gamma \cdot s\sqrt{m} \right\}.
\end{aligned}$$

Here, L_{YES} is a collection of N points at least one of which is close to the corresponding lattice, and L_{NO} is a collection of N points all of which are far from the corresponding lattices.

Consider the gap version of the *closest vector problem*, i.e., the language $L'_\gamma = \{L'_{YES}, L'_{NO}\}$ defined as:

$$L'_{YES} = \left\{ (\mathbf{B}, \mathbf{z}, t) \mid \exists \mathbf{s} \ : \ ||\mathbf{z} - \mathbf{B}^T \mathbf{s}|| \leq t \right\}.$$

$$L'_{NO} = \left\{ (\mathbf{B}, \mathbf{z}, t) \mid \forall \mathbf{s} \ : \ ||\mathbf{z} - \mathbf{B}^T \mathbf{s}|| > \gamma \cdot t \right\}.$$

It is known [20, 24] that there is an (interactive) witness-indistinguishable (WI) proof system for L'_γ when $\gamma \geq O(\sqrt{m/\log m})$. Observe that $L_{s,\gamma}$ can be described as the disjunction of several instance of L'_γ; that is,

$$\left(\begin{array}{c} \mathbf{B}_1, \ldots, \mathbf{B}_N \\ \mathbf{z}_1, \ldots, \mathbf{z}_N \end{array} \right) \in L_{YES} \Leftrightarrow \bigvee_i \left((\mathbf{B}_i, \mathbf{z}_i, s\sqrt{m}) \in L'_{YES} \right) .$$

$$\left(\begin{array}{c} \mathbf{B}_1, \ldots, \mathbf{B}_N \\ \mathbf{z}_1, \ldots, \mathbf{z}_N \end{array} \right) \in L_{NO} \Leftrightarrow \bigwedge_i \left((\mathbf{B}_i, \mathbf{z}_i, s\sqrt{m}) \in L'_{NO} \right) .$$

We can thus use the techniques of Cramer, Damgård, and Schoenmakers [17] to obtain an interactive WI proof for $L_{s,\gamma}$ with negligible soundness error. Using the Fiat-Shamir transformation [18], the resulting protocol can be made non-interactive in the random oracle model. These observations are summarized in the following lemma.

Lemma 6 *Let $\gamma \geq O(\sqrt{m/\log m})$. There is an NIWI proof system for $L_{s,\gamma}$ in the random oracle model, where the length of the proof is $O(mnN \log q)$ bits.*

We remark that for our application we only require soundness (and do not require the proof system to be a proof of knowledge) and witness indistinguishability (rather than zero knowledge).

3 A Group Signature Scheme Based on Lattices

3.1 Definitions

We adopt the definition of group signature schemes from the work of Bellare, Micciancio, and Warinschi [7], with the relaxation suggested by Boneh, Boyen, and Shacham [10] (and considered also in, e.g., [11]). Formally, a group signature scheme $\mathcal{GS} = (\mathsf{G.KeyGen}, \mathsf{G.Sign}, \mathsf{G.Vrfy}, \mathsf{G.Open})$ is a collection of four polynomial-time algorithms defined as follows.

- The *group key-generation algorithm* $\mathsf{G.KeyGen}(1^n, 1^N)$ is a randomized algorithm that takes a security parameter 1^n and the group size 1^N as input, and outputs $(\mathsf{PK}, \mathsf{TK}, \vec{\mathsf{gsk}})$, where PK is the group public key, TK is the group manager's tracing key, and $\vec{\mathsf{gsk}}$ is a vector of N signing keys with $\mathsf{gsk}[i]$ being the signing key given to the i^{th} group member.

- The *group signature algorithm* $\mathsf{G.Sign}(\mathsf{gsk}[i], M)$ is a randomized algorithm that takes as input a secret signing key $\mathsf{gsk}[i]$ and a message M, and outputs a signature σ.

- The *group signature verification algorithm* $\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma)$ is a deterministic algorithm that takes as input the group public key PK, a message M, and a signature σ, and outputs either 1 or 0 (signifying accept or reject, respectively).

- The *opening algorithm* $\mathsf{G.Open}(\mathsf{TK}, M, \sigma)$ is a deterministic algorithm that takes as input the tracing key TK, a message M, and a signature σ, and outputs an identity $i \in [N]$.

The basic consistency requirements of a group signature scheme are that an honest signature generated by a group member should be accepted as correct, and must be traceable to the group member who issued it. That is, for any $(\mathsf{PK}, \mathsf{TK}, \vec{\mathsf{gsk}})$ output by $\mathsf{G.KeyGen}(1^n, 1^N)$, any M, and any $i \in [N]$, if $\sigma \leftarrow \mathsf{G.Sign}(\mathsf{gsk}[i], M)$ then

$$\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma) = 1 \text{ and } \mathsf{G.Open}(\mathsf{TK}, M, \sigma) = i,$$

except with negligible probability over the entire experiment.

Group signature schemes are also required to satisfy two basic security properties: *anonymity* and *traceability.* Anonymity means that without the tracing key it should be infeasible to determine which group member issued a particular signature (even given all the signing keys). Bellare et al. [7] defined a "CCA-version" of this notion, where the adversary is given access to a tracing oracle. Following [10] we use a "CPA-version" of anonymity where such oracle access is not given.

Definition 1 *A group signature scheme* $\mathcal{GS} = (\mathsf{G.KeyGen}, \mathsf{G.Sign}, \mathsf{G.Vrfy}, \mathsf{G.Open})$ *is* anonymous *if for all polynomials $N(\cdot)$ and all probabilistic polynomial-time adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in n:*

1. *Compute* $(\mathsf{PK}, \mathsf{TK}, \vec{\mathsf{gsk}}) \leftarrow \mathsf{G.KeyGen}(1^n, 1^N)$ *and give* $(\mathsf{PK}, \vec{\mathsf{gsk}})$ *to $\mathcal{A}$.*
2. *$\mathcal{A}$ outputs two identities $i_0, i_1 \in [N]$, along with a message M. A random bit b is chosen, and $\mathcal{A}$ is given* $\mathsf{G.Sign}(\mathsf{gsk}[i_b], M)$*. Finally, $\mathcal{A}$ outputs a bit b'.*

$\mathcal{A}$ succeeds (denoted Succ*) if $b' = b$, and the advantage of $\mathcal{A}$ is* $\left|\Pr[\mathsf{Succ}] - \frac{1}{2}\right|$.

Traceability means that it should be infeasible for an adversary who corrupts some set of users $\mathcal{C}$ to output a valid signature that cannot be traced to some member of $\mathcal{C}$.

Definition 2 *A group signature scheme* $\mathcal{GS} = (\mathsf{G.KeyGen}, \mathsf{G.Sign}, \mathsf{G.Vrfy}, \mathsf{G.Open})$ *is* traceable *if for all polynomials $N(\cdot)$ and all probabilistic polynomial-time adversaries $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following experiment is negligible in n:*

1. *Compute* $(\mathsf{PK}, \mathsf{TK}, \vec{\mathsf{gsk}}) \leftarrow \mathsf{G.KeyGen}(1^n, 1^N)$ *and give* $(\mathsf{PK}, \mathsf{TK})$ *to $\mathcal{A}$.*
2. *$\mathcal{A}$ may query the following oracles adaptively and in any order:*
   - *A* $\mathsf{Corrupt}$ *oracle that on input $i \in [N]$ returns* $\mathsf{gsk}[i]$.
   - *A* Sign *oracle that on input i, M outputs* $\mathsf{G.Sign}(\mathsf{gsk}[i], M)$.

   Let $\mathcal{C}$ be the set of identities queried to $\mathsf{Corrupt}$.
3. *At some point, $\mathcal{A}$ outputs a message M and a signature σ.*

$\mathcal{A}$ succeeds if (1) $\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma) = 1$ *and (2)* $\mathsf{Sign}(i, M)$ *was never queried for $i \notin \mathcal{C}$, yet (3)* $\mathsf{G.Open}(\mathsf{TK}, M, \sigma) \notin \mathcal{C}$.

3.2 Our Construction

Let $q = \mathsf{poly}(n), m \geq 8n \log q$, and $s = \omega(\sqrt{n \log q \log m})$ be parameters of the system. We let $H : \{0,1\}^* \to \mathbb{Z}_q^n$ be a hash function, to be modeled as a random oracle. The group signature scheme is defined as follows:

$\mathsf{G.KeyGen}(1^n, 1^N)$: First compute $(\mathbf{B}_1, \mathbf{S}_1), \ldots, (\mathbf{B}_N, \mathbf{S}_N) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$ and then, for $1 \leq i \leq N$, compute $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i)$. Output $\mathsf{PK} = \Big((\mathbf{A}_i, \mathbf{B}_i)_{i=1}^N\Big)$ as the public key, $\mathsf{TK} = (\mathbf{S}_i)_{i=1}^N$ as the tracing key, and $\mathsf{gsk} = (\mathbf{T}_i)_{i=1}^N$ as the users' signing keys.

$\mathsf{G.Sign}(\mathsf{gsk}[j], M)$: To sign message M using secret key $\mathsf{gsk}[j] = \mathbf{T}_j$, choose random $r \leftarrow \{0,1\}^n$, set $\bar{M} = M \| r$, and then compute $\mathbf{h}_i = H(\bar{M} \| i)$ for $1 \leq i \leq N$. Then:

- Compute $\mathbf{e}_j \leftarrow \mathsf{GPVInvert}(\mathbf{A}_j, \mathbf{T}_j, s, \mathbf{h}_j)$.
- For $i \neq j$, choose $\mathbf{e}_i \in \mathbb{Z}_q^m$ uniformly subject to the condition that $\mathbf{A}_i \mathbf{e}_i = \mathbf{h}_i \pmod q$.

For all i, sample $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$ and compute $\mathbf{z}_i = \mathbf{B}_i^T \mathbf{s}_i + \mathbf{e}_i \pmod q \in \mathbb{Z}_q^m$. Finally, construct an NIWI proof π for the gap language $L_{s,\gamma}$ as discussed in Section 2.5 (and using the witness $(\mathbf{s}_i, i)$). Output the signature $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$.

$\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma)$: Parse the signature as $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$ and set $\bar{M} = M \| r$. Output 1 iff the proof π is correct, and $\mathbf{A}_i \mathbf{z}_i = H(\bar{M} \| i) \pmod q$ for all i.

$\mathsf{G.Open}(\mathsf{TK}, M, \sigma)$: Parse the signature as $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$. Using the $\{\mathbf{S}_i\}$, output the smallest index i for which[4] $\mathsf{dist}(\Lambda(\mathbf{B}_i^T), \mathbf{z}_i) \leq s\sqrt{m}$.

We first check correctness. Let $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$ be a signature produced by an honest signer. It is clear that π is a valid proof. Moreover, for any i we have

$$\mathbf{A}_i \mathbf{z}_i = \mathbf{A}_i(\mathbf{B}_i^T \mathbf{s}_i + \mathbf{e}_i) = \mathbf{A}_i \mathbf{e}_i = H(\bar{M} \| i) \pmod q,$$

and so verification succeeds. Correctness of the opening algorithm follows easily.

Theorem 1 *Let m, q, and s be as described above. If $\mathsf{LWE}_{m,q,\alpha}$ is hard for $\alpha = s/(q\sqrt{2})$, and the proof system used is witness indistinguishable, then the group signature scheme described above is anonymous. If GapSVP_γ is hard for $\gamma = O(n \log^4 n)$, then the group signature scheme described above is traceable.*

We note that for values of s as described above, the hardness of $\mathsf{LWE}_{m,q,\alpha}$ is implied by the difficulty of finding a quantum algorithm for approximating $\mathsf{GapSVP}_{\hat{\gamma}}$ for $\hat{\gamma} = \widetilde{O}(n/\alpha)$ [28], so our entire scheme can be based on the difficulty of finding a quantum algorithm for GapSVP.

We prove anonymity in Section 3.3 and traceability in Section 3.4.

[4] Soundness of the proof system ensures that if σ is valid, then some such i exists except with negligible probability.

3.3 Anonymity

Fix $N = \mathsf{poly}(n)$ and let $\mathcal{A}$ be a PPT adversary attacking the group signature scheme in the sense of Definition 1. Let G_0 denote the experiment of Definition 1 with $b = 0$, and let G_1 be the same experiment with $b = 1$. We consider a sequence of experiments $\mathsf{G}_0, \mathsf{G}'_0, \mathsf{G}'_1, \mathsf{G}_1$ and show that each experiment is indistinguishable from the one preceding it. This implies anonymity.

We review G_0 as applied to our group signature scheme. First, the key-generation algorithm $\mathsf{G.KeyGen}(1^n, 1^N)$ is run and $\mathcal{A}$ is given the public key $\mathsf{PK} = \Big((\mathbf{A}_i, \mathbf{B}_i)_{i=1}^N\Big)$ and the users' secret keys $\mathsf{gsk} = (\mathbf{T}_i)_{i=1}^N$, where each $\mathbf{B}_i$ is statistically close to uniform and $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i)$. (The tracing key TK is irrelevant in the CPA-version of the anonymity experiment that we are considering.) Next, $\mathcal{A}$ outputs i_0, i_1, M, and is given a signature of user i_0 on M, computed as follows. Let $\mathbf{h}_i = H(M\|r\|i)$, for a random $r \in \{0,1\}^n$. Then $\mathbf{e}_{i_0}$ is computed as $\mathbf{e}_{i_0} \leftarrow \mathsf{GPVInvert}(\mathbf{A}_{i_0}, \mathbf{T}_{i_0}, s, \mathbf{h}_{i_0})$, whereas $\mathbf{e}_i$ (for $i \neq i_0$) is chosen uniformly subject to the condition that $\mathbf{A}_i\mathbf{e}_i = \mathbf{h}_i \pmod q$. Then, for all $i \in [N]$, choose random $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$ and compute $\mathbf{z}_i = \mathbf{B}_i^T\mathbf{s}_i + \mathbf{e}_i$. Finally, a proof π is generated and $\mathcal{A}$ is given the signature $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$.

In G'_0 we introduce the following modification with respect to G_0: when generating the signature, we now compute $\mathbf{e}_{i_0} \leftarrow \mathsf{GPVInvert}(\mathbf{A}_{i_0}, \mathbf{T}_{i_0}, s, \mathbf{h}_{i_0})$ and $\mathbf{e}_{i_1} \leftarrow \mathsf{GPVInvert}(\mathbf{A}_{i_1}, \mathbf{T}_{i_1}, s, \mathbf{h}_{i_1})$. (For $j \notin \{i_0, i_1\}$, the value $\mathbf{e}_j$ is computed as before.)

Claim 1 *If the* $\mathsf{LWE}_{m,q,\alpha}$ *problem is hard, then* G_0 *and* G'_0 *are computationally indistinguishable.*

Proof Recall (cf. Lemma 2) that hardness of the $\mathsf{LWE}_{m,q,\alpha}$ problem implies hardness of the $\widehat{\mathsf{LWE}}_{m,q,\alpha q\sqrt{2}}$ problem. We use $\mathcal{A}$ to construct a PPT algorithm $\mathcal{D}$ for the $\widehat{\mathsf{LWE}}_{m,q,\alpha q\sqrt{2}}$ problem. $\mathcal{D}$ is given as input $(\mathbf{B}, \mathbf{y}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, where $\mathbf{B}$ is uniform and $\mathbf{y}$ is either uniform or equal to $\mathbf{B}^T\mathbf{s} + \mathbf{e}$ for $\mathbf{e} \sim D_{\mathbb{Z}^m, \alpha q\sqrt{2}}$.

$\mathcal{D}$ first chooses a random index $i^* \leftarrow [N]$ and sets $\mathbf{B}_{i^*} = \mathbf{B}$. For all $i \neq i^*$, it chooses $\mathbf{B}_i$ uniformly at random. Then, for $1 \le i \le N$ algorithm $\mathcal{D}$ computes $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i)$. It gives $\mathsf{PK} = \Big((\mathbf{A}_i, \mathbf{B}_i)_{i=1}^N\Big)$ and $\mathsf{gsk} = (\mathbf{T}_i)_{i=1}^N$ to $\mathcal{A}$. All H-queries of $\mathcal{A}$ are answered with random elements from the appropriate domain.

Eventually $\mathcal{A}$ outputs two identities $i_0, i_1 \in [N]$ along with a message M. If $i^* \neq i_1$ then $\mathcal{D}$ outputs a random bit and aborts. Otherwise, $\mathcal{D}$ creates a signature by choosing random $r \in \{0,1\}^n$ and fixing[5] $\mathbf{h}_{i_1} \overset{\text{def}}{=} H(M\|r\|i_1) = A_{i_1}\mathbf{y}$. (The value $\mathbf{h}_i = H(M\|r\|i)$ for $i \neq i_1$ is chosen uniformly.) Then $\mathcal{D}$ computes $\mathbf{e}_{i_0} \leftarrow \mathsf{GPVInvert}(\mathbf{A}_{i_0}, \mathbf{T}_{i_0}, s, \mathbf{h}_{i_0})$ and, for $i \notin \{i_0, i_1\}$, chooses $\mathbf{e}_i$ uniformly subject to the condition that $\mathbf{A}_i\mathbf{e}_i = \mathbf{h}_i \pmod q$. ($\mathcal{D}$ does not explicitly compute any value $\mathbf{e}_{i_1}$.) For $i \neq i_1$, the ciphertext $\mathbf{z}_i$ is computed as in G_0 and G'_0. However, $\mathcal{D}$ sets $\mathbf{z}_{i_1} = \mathbf{y}$.

Let $\mathcal{D}_{\text{rand}}$ denote the above experiment when $\mathcal{D}$'s input $\mathbf{y}$ is uniformly distributed. We claim that $\mathcal{A}$'s view in $\mathcal{D}_{\text{rand}}$ is statistically close to its view in G_0. Indeed:

- In G_0 we have $\mathbf{h}_{i_1}$ chosen uniformly in $\mathbb{Z}_q^n$; then $\mathbf{e}_{i_1}$ is chosen uniformly subject to $\mathbf{A}_{i_1}\mathbf{e}_{i_1} = \mathbf{h}_{i_1}$; and finally $\mathbf{z}_{i_1} = \mathbf{B}_{i_1}^T\mathbf{s}_{i_1} + \mathbf{e}_{i_1}$.
- In $\mathcal{D}_{\text{rand}}$ we have $\mathbf{z}_{i_1} = \mathbf{y} = \mathbf{B}_{i_1}^T\mathbf{s}_{i_1} + \mathbf{e}_{i_1}$ for $\mathbf{e}_{i_1}$ chosen uniformly in $\mathbb{Z}_q^m$; then $\mathbf{h}_{i_1} = \mathbf{A}_{i_1}\mathbf{e}_{i_1}$.

To see that these are statistically close, we demonstrate that the choice of $\mathbf{e}_{i_1}$ in G_0 is statistically close to uniform over $\mathbb{Z}_q^m$. We view $\mathbf{A}$ as a function from $\mathbb{Z}_q^m \to \mathbb{Z}_q^n$, and note that this function is

[5] Note that, except with negligible probability, $H(M\|r\|i_1)$ has not been queried thus far.

regular. Furthermore, since the columns of $\mathbf{A}$ generate all of $\mathbb{Z}_q^n$ with all but negligible probability (over the choice of $\mathbf{A}$), our randomly chosen $\mathbf{h}$ is uniform over the image of $\mathbf{A}$. For a regular function, choosing a uniform element from the image, followed by a uniform element from its pre-image, is equivalent to choosing a uniform element from the domain, as is done in $\mathcal{D}_{\text{rand}}$.

On the other hand, let $\mathcal{D}_{\text{LWE}}$ denote the above experiment when $\mathcal{D}$'s input $\mathbf{y}$ is distributed according to $\mathbf{y} = \mathbf{B}^T\mathbf{s} + \mathbf{e}$ for $\mathbf{e} \sim D_{\mathbb{Z}^m, \alpha q\sqrt{2}}$. We claim that $\mathcal{A}$'s view in $\mathcal{D}_{LWE}$ is statistically close to its view in G_0'. Indeed:

- In experiment G_0' we have $\mathbf{h}_{i_1}$ chosen uniformly in $\mathbb{Z}_q^n$. Next, we compute $\mathbf{e}_{i_1} \leftarrow \mathsf{GPVInvert}(\mathbf{A}_{i_1}, \mathbf{T}_{i_1}, s, \mathbf{h}_{i_1})$; and finally $\mathbf{z}_{i_1} = \mathbf{B}_{i_1}^T\mathbf{s}_{i_1} + \mathbf{e}_{i_1}$.
- In $\mathcal{D}_{LWE}$ we have $\mathbf{z}_{i_1} = \mathbf{y} = \mathbf{B}_{i_1}^T\mathbf{s}_{i_1} + \mathbf{e}_{i_1}$ for $\mathbf{e}_{i_1} \sim D_{\mathbb{Z}^m, \alpha \cdot q \cdot \sqrt{2}}$; then $\mathbf{h}_{i_1} = \mathbf{A}_{i_1}\mathbf{e}_{i_1}$.

The above are easily seen to be statistically close for our choice of parameters, again using the results of [19]. Since the probability that $\mathcal{D}$ does not abort is $1/N$, and its decision to abort is independent of $\mathcal{A}$'s success, the proof is complete. ■

The rest of the proof of anonymity is straightforward, and so we merely provide a sketch. Experiment G_1' is identical to G_0' with the exception that the proof π is now computed using the witness $(\mathbf{s}_{i_1}, i_1)$ rather than $(\mathbf{s}_{i_0}, i_0)$. Witness indistinguishability of the proof system implies that G_1' and G_0' are computationally indistinguishable.

Computational indistinguishability of G_1' and G_1 (the experiment from Definition 1 with $b = 1$) can be proved exactly as in the proof o the previous claim.

3.4 Traceability

Fix $N = \mathsf{poly}(n)$ and let $\mathcal{A}$ be a PPT adversary attacking the group signature scheme in the sense of Definition 2. We construct a PPT forger $\mathcal{F}$ for the GPV signature scheme [19] (in the random oracle model) whose success probability is polynomially related to that of $\mathcal{A}$. Since the GPV signature scheme is secure assuming hardness of the GapSVP_γ problem, this completes the proof.

We first observe that we may, without loss of generality, assume that $\mathcal{A}$ never corrupts *all* users in $[N]$ because $\mathcal{A}$ can succeed with only negligible probability in this case. (Given a valid signature $(r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$, soundness of the proof system implies that $\mathsf{G.Open}$ outputs some $i \in [N]$ except with negligible probability.) We will assume this in what follows.

$\mathcal{F}$ is given a public key $\mathbf{A}$ for the GPV signature scheme, and begins by choosing a random index $i^* \in [N]$ and setting $\mathbf{A}_{i^*} = \mathbf{A}$. Next, it computes the values $(\mathbf{B}_{i^*}, \mathbf{S}_{i^*}) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_{i^*})$. For all the remaining indices $i \neq i^*$, the forger computes the values $(\mathbf{B}_i, \mathbf{S}_i) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$ and $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i)$ exactly as in the legitimate key-generation algorithm. After this, $\mathcal{F}$ gives $\mathsf{PK} = (\mathbf{A}_i, \mathbf{B}_i)_{i=1}^N$ and $\mathsf{TK} = (\mathbf{S}_i)_{i=1}^N$ to $\mathcal{A}$. We note that by Corollary **??**, the distribution of these keys is statistically close to the distribution that is expected by the adversary.

$\mathcal{F}$ answers random oracle queries of $\mathcal{A}$ by simply passing these queries to its own random oracle. $\mathcal{F}$ responds to the other queries of $\mathcal{A}$ as follows:

- $\mathsf{Corrupt}(i)$: if $i \neq i^*$ then $\mathcal{F}$ gives $\mathbf{T}_i$ to $\mathcal{A}$, while if $i = i^*$ then $\mathcal{F}$ aborts.
- $\mathsf{Sign}(i, M)$: If $i \neq i^*$ then $\mathcal{F}$ computes the signature using $\mathbf{T}_i$ and the honest signing algorithm. If $i = i^*$, then:

1. $\mathcal{F}$ chooses random $r \in \{0,1\}^n$ and queries its own signing oracle on the message $M\|r\|i^*$. It receives in return a signature $\mathbf{e}_{i^*}$.
2. The remainder of the signature is computed using the honest signing algorithm. (Note that computation of $\mathbf{e}_{i^*}$ the only aspect of signing that relies on the secret key of user i^*.)

Let $\mathcal{C}$ denote the set of identities that $\mathcal{A}$ has queried to $\mathsf{Corrupt}$. (Recall that if $\mathcal{F}$ has not aborted, then $i^* \notin \mathcal{C}$.) At some point $\mathcal{A}$ outputs a message M and signature $\sigma = (r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$. Assume $\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma) = 1$, and that $\mathsf{Sign}(i, M)$ was never queried for $i \notin \mathcal{C}$. Since $\mathcal{F}$ has the tracing key TK, it can compute $j \leftarrow \mathsf{G.Open}(\mathsf{TK}, M, \sigma)$. If $j \neq i^*$ then $\mathcal{F}$ aborts. Otherwise, $\mathcal{F}$ does:

1. Use $\mathbf{S}_{i^*}$ to recover $\mathbf{e}_{i^*}$ such that
   - $||\mathbf{e}_{i^*}||_\infty \leq s\sqrt{m}$, and
   - $\mathbf{z}_{i^*} - \mathbf{e}_{i^*} \in \mathcal{L}(\mathbf{B}_{i^*}^T)$.
2. Output the forgery $(M\|r\|i^*, \mathbf{e}_{i^*})$.

Let ϵ denote the probability with which $\mathcal{A}$ succeeds in the experiment of Definition 2. It is easy to see that $\mathcal{F}$ aborts with probability at most[6] $(N-1)/N$ and, conditioned on not aborting, the view of $\mathcal{A}$ when run as a sub-routine by $\mathcal{F}$ is statistically close to its view in the experiment described in Definition 2. Thus, with probability at least ϵ/N it holds that $\mathcal{A}$ outputs (M, σ) with $\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma) = 1$ and $\mathsf{G.Open}(\mathsf{TK}, M, \sigma) = i^*$, and where $\mathcal{A}$ never queried $\mathsf{Sign}(i^*, M)$. We show that whenever this occurs, then $\mathcal{F}$ outputs a valid forgery (except with negligible probability).

Fix (M, σ) such that the above hold, and let $\sigma = (r, \mathbf{z}_1, \ldots, \mathbf{z}_N, \pi)$. Since $\mathsf{G.Open}(\mathsf{TK}, M, \sigma) = i^*$, this implies that $\mathcal{F}$ will indeed be able to recover $\mathbf{e}_{i^*}$ such that (1) $||\mathbf{e}_{i^*}||_\infty \leq s\sqrt{m}$ and (2) $\mathbf{z}_{i^*} - \mathbf{e}_{i^*} \in \mathcal{L}(\mathbf{B}_{i^*}^T)$. Moreover, since $\mathsf{G.Vrfy}(\mathsf{PK}, M, \sigma) = 1$ we have $\mathbf{A}_{i^*}\mathbf{z}_{i^*} = H(M\|r\|i^*)$; since $\mathbf{A}_{i^*}(\mathbf{z}_{i^*} - \mathbf{e}_{i^*}) = \mathbf{0}$ this means $\mathbf{A}_{i^*}\mathbf{e}_{i^*} = H(M\|r\|i^*)$. Thus $\mathbf{e}_{i^*}$ is a valid GPV signature on the message $M\|r\|i^*$. Since $\mathcal{A}$ never queried $\mathsf{Sign}(i^*, M)$, we know that $\mathcal{F}$ never queried its own signing oracle for a signature on $M\|r\|i^*$. It follows that the output of $\mathcal{F}$ is indeed a valid forgery.

Acknowledgments. We thank Chris Peikert for pointing out that the results from [26] can be used to prove Lemma 2, and the anonymous reviewers of Asiacrypt 2010 for helpful comments.

References

[1] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 99–108. ACM Press, May 1996.

[2] M. Ajtai. Generating hard instances of the short basis problem. In *26th Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.

[3] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 75–86. Schloss Dagstuhl, 2009. Available from `http://drops.dagstuhl.de`. Full version to appear in *Theory of Computing Systems.*

[6] Actually, $\mathcal{F}$ aborts with probability at most $(N-1)/N + \mathsf{negl}(n)$, where the negligible term arises from the possibility that $\mathcal{A}$ violates soundness of the proof system. We ignore this for simplicity.

[4] G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros. Practical group signatures without random oracles, 2005. Cryptology ePrint Archive, report 2005/385.

[5] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology — Crypto 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.

[6] G. Ateniese, D. X. Song, and G. Tsudik. Quasi-efficient revocation in group signatures. In *Financial Cryptography and Data Security 2002*, volume 2357 of *LNCS*, pages 183–197. Springer, 2002.

[7] M. Bellare, D. Micciancio, and B. 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 *LNCS*, pages 614–629. Springer, 2003.

[8] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conf. on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[9] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Cryptographers' Track — RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.

[10] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Advances in Cryptology — Crypto 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[11] X. Boyen and B. Waters. Compact group signatures without random oracles. In *Advances in Cryptology — Eurocrypt 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006.

[12] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *10th Intl. Conference on Theory and Practice of Public Key Cryptography(PKC 2007)*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.

[13] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — Crypto 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.

[14] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology — Crypto 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004.

[15] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology — Eurocrypt 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010.

[16] D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology — Eurocrypt '91*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.

[17] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology — Crypto '94*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.

[18] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — Crypto '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.

[19] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 197–206. ACM Press, 2008.

[20] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Computer and System Sciences*, 60(3):540–563, 2000.

[21] J. Groth. Fully anonymous group signatures without random oracles. In *Advances in Cryptology — Asiacrypt 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.

[22] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 198–214. Springer, 2005.

[23] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[24] D. Micciancio and S. P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Advances in Cryptology — Crypto 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, 2003.

[25] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 333–342. ACM Press, 2009.

[26] C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *Advances in Cryptology — Crypto 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010.

[27] C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 536–553. Springer, 2008.

[28] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.