Revisiting Preimage Sampling for Lattices

Corentin Jeudy[1,2], Adeline Roux-Langlois[3], and Olivier Sanders[1]

corentin.jeudy@orange.com, adeline.roux-langlois@cnrs.fr,
olivier.sanders@orange.com

[1] Orange Labs, Applied Crypto Group, Cesson-Sévigné, France
[2] Univ Rennes, CNRS, IRISA, Rennes, France
[3] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Abstract. Preimage Sampling is a fundamental process in lattice-based cryptography whose performance directly affects the one of the cryptographic mechanisms that rely on it. In 2012, Micciancio and Peikert proposed a new way of generating trapdoors (and an associated preimage sampling procedure) with very interesting features. Unfortunately, in some applications such as digital signatures, the performance may not be as competitive as other approaches like Fiat-Shamir with Aborts.
In this work we revisit the Micciancio-Peikert preimage sampling algorithm with different contributions. We first propose a finer analysis of this procedure which results in interesting efficiency gains of around 20% on the preimage sizes without affecting security. It can thus be used as a drop-in replacement in every construction resorting to it.
We then reconsider the Lyubashevsky-Wichs sampler for Micciancio-Peikert trapdoors which leverages rejection sampling but suffered from strong parameter requirements that hampered performance. We propose an improved analysis which allows to obtain much more compact parameters. This leads to gains of up to 30% compared to the original Micciancio-Peikert sampling technique and opens promising perspectives for the efficiency of advanced lattice-based constructions relying on such mechanisms.
As an application of the latter, we give the first lattice-based aggregate signature supporting public aggregation and that achieves relevant compression compared to the concatenation of individual signatures. Our scheme is proven secure in the aggregate chosen-key model coined by Boneh et al. in 2003, based on the well-studied assumptions Module Learning With Errors and Module Short Integer Solution.

Keywords: Lattice-Based Cryptography · Trapdoors · Preimage Sampling · Aggregate Signature

1 Introduction

Lattice-based cryptography has proven to be a relatively stable and extensively studied candidate to provide post-quantum secure primitives, and has now shifted towards proposing concretely efficient constructions. The NIST standardization [NIS] perfectly reflects this trend as they recently announced the

first round of future standards, which is dominated by lattice-based constructions [BDK+18,DKL+18,PFH+20], and are moving to practical deployment discussions. The versatility of lattice-based cryptography has also given rise to more advanced constructions, but that are not yet represented in standardization efforts due to their remaining efficiency gap compared to currently deployed pre-quantum[4] solutions. Typically, despite very recent results, e.g., [dPK22,LNP22], lattice-based blind signatures and group signatures still yield signatures that are about 1000 times larger than their pre-quantum counterparts and are thus unlikely to be included the corresponding ISO/IEC standards [ISO13,ISO16] at this stage. Improving the performance of such primitives is therefore paramount before considering standardization and integration. For that, it seems necessary to propose new techniques and to reassess some widely adopted techniques in order to identify their limitations and possibly some margin for optimization.

This work is mostly directed at the realm of lattice-based signatures, but it may find applications in other areas of lattice cryptography. Lattice-based signature schemes are usually designed by following one of two main paradigms. The first one, called the *hash-and-sign* paradigm, was instantiated by Gentry et al. [GPV08] with lattice preimage sampleable trapdoor functions. In such schemes, the signing key consists of a trapdoor for a publicly computable function which allows to efficiently find short preimages. Signatures are then preimages of seemingly random (and possibly message-dependent) syndromes. Only the signer is able to compute such preimages, but everyone is able to compute the image to ensure they represent valid signatures. Several schemes rely on variants of the above, e.g., [GPV08,MP12,DM14,DLP14], and were successfully pushed towards concrete practicality [PFH+20,EFG+22] using an additional assumption. Trapdoor preimage sampleable functions also represent the most widely used building block in the design of more advanced forms of signatures such as group signatures [dPLS18,LNPS21], blind signatures [AKSY22,dPK22], signatures with efficient protocols [LLM+16,JRS22], etc. In their general use, trapdoor preimage sampling can however be quite computationally intensive and most preimage sampling algorithms are designed to only support Gaussian-distributed preimages.

An alternative, called the *Fiat-Shamir with Aborts* (FSwA) paradigm, was proposed by Lyubashevsky [Lyu12], building signatures on Schnorr-like proofs made non-interactive with the Fiat-Shamir transform. This framework avoids the use of trapdoors, and uses rejection sampling to control the distribution of signatures while making them independent of the signing key. Even though most applications yield Gaussian-distributed signatures, it is possible to tweak the rejection sampling step to get other distributions that can be more suitable depending on the context. Efficient instantiations of this signature paradigm were proposed, such as qTESLA [ABB+20] and Dilithium [DKL+18].

Interestingly, in [LW15], Lyubashevsky and Wichs show that these two approaches may be combined in the case of Micciancio-Peikert trapdoors [MP12].

[4] We use pre-quantum to refer to cryptography that does not withstand the power of quantum computing.

They indeed propose for the latter a new sampling method relying on rejection sampling. This method allows for a broader choice of preimage distributions but unfortunately suffers from parameters requirements that, in the end, makes it less efficient than the original sampling method.

1.1 Our Contributions

In this paper, we focus on improving the preimage sampler associated to the trapdoor functions from [MP12], which is the core of many advanced lattice constructions, e.g., [DM14,BFRS18,dPLS18,BEP+21,LNPS21,LNP22,dPK22,JRS22]. It also has nice connections with other approaches as illustrated by the result in [LW15]. We first propose a finer analysis of the existing procedure resulting in drastic gains without affecting its security. On the contrary, it leads to slightly enhanced security guarantees and can thus be used as a drop-in replacement for every lattice constructions using preimage sampling based on the trapdoor functions of [MP12]. We then revisit the approach in [LW15] to show that we can significantly alleviate the requirements in the original security analysis, at least in case of the most common applications of preimage sampling. It entails dramatic gains in performance and thus enables the approach from Lyubashevsky and Wichs to achieve its full potential. We note that these contributions apply to constructions on both standard and structured lattices. Finally, we show that our new preimage sampling procedure unlocks the design of new constructions on lattices that only existed in the pre-quantum world prior to our work. More specifically, we propose the first lattice-based aggregate signature scheme that supports public aggregation and that has relevant compression rates with respect to simply concatenating individual signatures.

Starting Point. In [MP12], Micciancio and Peikert propose a preimage sampling algorithm for matrices $\mathbf{A_H} = [\mathbf{A}|\mathbf{HG} - \mathbf{AR}]$, where $\mathbf{R}$ constitutes the trapdoor. More precisely, $\mathbf{A}$ is uniform matrix in $\mathbb{Z}_q^{d\times 2d}$, $\mathbf{H}$ is a tag matrix in $GL_d(\mathbb{Z}_q)$, $\mathbf{G} \in \mathbb{Z}^{d\times kd}$ (with $k = \log_2 q$) is the base-b gadget matrix introduced in [MP12], and $\mathbf{R}$ is a short matrix. Their algorithm uses the knowledge of $\mathbf{R}$ to sample $\mathbf{v} \in \mathbb{Z}^{(2+k)d}$ according to a spherical discrete Gaussian of parameter σ such that $\mathbf{A_H v} = \mathbf{u} \bmod q$ for an input syndrome $\mathbf{u}$. The technique first relies on the observation that if $\mathbf{z}$ is a Gaussian with width $\sigma_{\mathbf{G}}$ such that $\mathbf{HGz} = \mathbf{u}$, then the vector $\mathbf{v'} = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$ is a valid candidate. This naive approach leaks information on the trapdoor $\mathbf{R}$, which is why the authors perturb this solution $\mathbf{v'}$ into $\mathbf{v} = \mathbf{p} + \mathbf{v'}$, for some suitable perturbation vector $\mathbf{p}$, while adjusting $\mathbf{z}$ to verify $\mathbf{HGz} = \mathbf{u} - \mathbf{A_H p}$. By carefully choosing the covariance of the Gaussian $\mathbf{p}$, one can indeed ensure that $\mathbf{v}$ follows a spherical Gaussian distribution of width σ, which in turn does not leak information on the trapdoor.

Contribution 1: From Spherical to Elliptical. Our first contribution consists in a finer analysis of the approach above. We indeed observe that the information on $\mathbf{R}$ in $\mathbf{v'} = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$ is symmetrically drowned by $\mathbf{p}$ to ob-

tain a spherical distribution, which results in a Gaussian $\mathbf{v}$ with parameter $\sigma = \Theta(\sigma_{\mathbf{G}} \cdot \|\mathbf{R}\|_2)$. This is not optimal as $\mathbf{v}'_1 = \mathbf{Rz}$ and $\mathbf{v}'_2 = \mathbf{z}$ are not of the same size, in particular for small bases b, and as $\mathbf{v}_1$ is the only one depending on the secret $\mathbf{R}$.

A first attempt to break the symmetry could be to only perturb the first part $\mathbf{v}_1$ but the result is insecure, as we explain in Section 3.1. We then revisit the original security analysis based on the convolution theorem by considering different widths σ_1 and σ_2 for $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{v}'_1$ and $\mathbf{v}_2 = \mathbf{p}_2 + \mathbf{v}'_2$, with the goal of decreasing σ_2 as much as possible while retaining the same security level. This approach is indeed particularly relevant when recalling that $\mathbf{v}'_2$ does not depend on $\mathbf{R}$ and therefore does not need to be perturbed as much as $\mathbf{v}'_1$. More concretely, we show that we can use $\sigma_1 = \Theta(\sigma_{\mathbf{G}} \cdot \|\mathbf{R}\|_2)$ with the same constant up to a $\sqrt{2}$ factor, but $\sigma_2 = \sigma_1 / \|\mathbf{R}\|_2$. It thus allows us to keep $\mathbf{v}_1$ (almost) as before while dramatically reducing the size of $\mathbf{v}_2$.

This modification alone reduces the bit-size of $\mathbf{v}$ between 20 and 30%. Additionally, because $\mathbf{v}_1$ has roughly the same size, it also improves the expected Euclidean norm $\|\mathbf{v}\|_2$ which usually leads to increased security. We thus gain on all metrics and are conceptually close to the original method, meaning our result can be used as a drop-in replacement in every primitive using such preimage sampling.

We note that this approach is different from the recent technique proposed by Espitau et al. [ETWY22] in the context of compressing hash-and-sign signatures. Indeed, when moving from spherical to elliptical Gaussians, they shrink the part of the preimage that corresponds to the outputted signature, but expand by the same factor the part of the preimage that is recovered during verification. Their optimization applies to hash-and-sign signatures that rely on different preimage sampling procedures, such as [PFH+20,EFG+22], which are not gadget-based as that of [MP12].

Contribution 2: Re-assessing the Lyubashevsky-Wichs Sampler. Although we managed to improve for free the efficiency of preimage sampling, it remains quite rigid as it requires sampling perturbations $\mathbf{p}$ from highly non-spherical Gaussian, and is limited to Gaussian preimages. To circumvent these limitations, we revisit the approach from Lyubashevsky and Wichs [LW15] that further breaks the symmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$. Their idea is to set $\mathbf{p}_2 = \mathbf{0}$ and $\mathbf{z} = \mathbf{G}^{-1}(\mathbf{u} - \mathbf{Ap}_1)$ where $\mathbf{G}^{-1}(\cdot)$ is the base-b decomposition. Directly outputting $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ and $\mathbf{v}_2 = \mathbf{z}$ again leaks information on $\mathbf{R}$ because of $\mathbf{v}_1$ and they thus need to adjust this approach. Actually, by identifying $\mathbf{Ap}_1$, $\mathbf{z}$ and $\mathbf{v}_1$ with (respectively) the commitment, the challenge and the answer of a zero-knowledge proof of knowledge of $\mathbf{R}$, this problem is very similar to the one of Fiat-Shamir signature in [Lyu12]. They then resort to the same workaround, namely rejection sampling: before outputting $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ and $\mathbf{v}_2 = \mathbf{z}$, one performs rejection sampling on $\mathbf{v}_1$ to make its distribution independent of $\mathbf{R}$ and $\mathbf{z}$.

However, to thoroughly show that the preimages do not leak information on $\mathbf{R}$, they provide a simulation result which suffers from parameter constraints that makes it less efficient than the original sampler from [MP12] in terms of preimage size. More concretely, they show that the output distribution of the preimages is statistically close to a distribution that does not depend on the trapdoor $\mathbf{R}$ for an arbitrary (potentially adversarial) syndrome $\mathbf{u}$. Because they deal with an arbitrary $\mathbf{u}$, nothing can be assumed about its distribution which in turn places strong restrictions on the parameters to compensate. Indeed, in their result, they need to assume that $\mathbf{A}\mathbf{v}_1$ (and $\mathbf{A}\mathbf{p}_1$) is *statistically* close to uniform requiring the parameters to be large in order to use a regularity lemma. This requirement in turn prevents them from using a computational instantiation of MP trapdoors. Since computational MP trapdoors lead to much smaller preimages, they are usually more compact than the ones generated by the sampler of [LW15].

Our second contribution is then to provide a improved analysis of the sampler from [LW15] to get rid of these restrictive requirements and thus obtain more compact preimages. In many situations in cryptography, the syndrome follows a prescribed distribution. For GPV signatures [GPV08] for example, the syndrome $\mathbf{u}$ is the hash output of the message $\mathcal{H}(\mathbf{m})$ where $\mathcal{H}$ is modeled as a random oracle. This means that the syndrome we expect are uniformly distributed and cannot be controlled by the adversary. Assuming the uniform distribution of $\mathbf{u}$ allows us to remove this constraint on $\mathbf{A}\mathbf{v}_1$ being statistically close to uniform, as we can, at a high level, use the randomness of $\mathbf{u}$ to achieve the same conclusions. As we show in our paper, removing this constraint removes the need for a large perturbation (either in norm or dimension) and thus leads to improved performances.

We give a comparison with the results of the first contribution by forcing a Gaussian distribution on $\mathbf{v}_1$. In this case, $\mathbf{p}_1$ must be drawn from a wide enough Gaussian with parameter $\sigma = \Theta(\|\mathbf{R}\mathbf{z}\|_2)$. Because $\mathbf{z}$ is the output of $\mathbf{G}^{-1}(\cdot)$, its infinity norm is bounded by b, which yields $\sigma = \Theta(\|\mathbf{R}\|_2 b\sqrt{kd})$. As opposed to the previous improvement, the size of $\mathbf{v}_1$ increases compared to [MP12], but $\mathbf{v}_2$ is now in base b which is can be much smaller (even minimal when $b = 2$ for example). For a GPV signature [GPV08] using the trapdoors from [MP12], the sampler from [LW15] with our improved simulation result actually decreases the bit-size of the overall signature, i.e., the total bit-size of $\mathbf{v}$, by 10% compared to Contribution 1 and thus by 30% compared to the original sampling method. The overall bit-size of said signatures drops below 8.5 KB, which shows promising perspectives for the efficiency of advanced lattice-based signatures using the trapdoors from [MP12].

Contribution 3: Application to Aggregate Signatures. As an example application of our new analysis of the sampler from [LW15], we propose an aggregate signature scheme based on structured lattices that fully leverages the asymmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$. An aggregate signature is a regular signature scheme completed by a mechanism $\mathsf{AggSign}$ taking the public keys pk_i of N users as well as pairs of message-signature $(\mathbf{m}_i, \mathsf{sig}_i)$ from each user, and compresses all

the sig_i into a single signature $\mathsf{sig}_{\mathsf{agg}}$. A second mechanism $\mathsf{AggVerify}$ is appended to verify that $\mathsf{sig}_{\mathsf{agg}}$ is a valid *aggregate* signature on the messages $\mathbf{m}_i$ under the keys pk_i, but without requiring the individual sig_i. One of the key features is that the aggregation is public and non-interactive, meaning it does not require the signers' secret keys nor does it need them to interact to produce $\mathsf{sig}_{\mathsf{agg}}$. A basic efficiency requirement is that the size of $\mathsf{sig}_{\mathsf{agg}}$ should be lower than the concatenation of the sig_i, the latter being the simplest form of aggregate signature.

Such primitives were first introduced by Boneh et al. [BGLS03], which has led to several efficient constructions on classical groups, such as for example the works in [BGLS03,BNN07,RS13,HKW15,HW18]. Post-quantum constructions were however unknown until the first attempt of Döroz et al. [DHSS20]. This lattice-based proposal turned out to be either less efficient than the trivial concatenation of signatures, or prone to attacks due to their compression technique as pointed out by Boudgoust and Roux-Langlois [BR21]. Additionally, their construction was based on a non-standard assumption called the Partial Fourier Recovery problem for which the hardness confidence is limited due to recent results by Boudgoust, Gachon and Pellet-Mary [BGP22]. Boudgoust and Roux-Langlois also proposed in [BR21] an aggregate signature based on module lattices following the FSwA signature paradigm. Again, it turned out that the peculiarities of aggregate signature security led to $\mathsf{sig}_{\mathsf{agg}}$ being larger than the concatenation.

In this work, we construct the first lattice-based aggregate signature with public aggregation that achieves relevant compression compared to the concatenation of individual signatures. Our scheme stems from the GPV signature [GPV08] instantiated with MP trapdoors [MP12], the sampler from [LW15] in our improved parameter setting as a key element. At a high level, each users has a key pair $(\mathsf{sk}_i, \mathsf{pk}_i) = (\mathbf{R}_i, \mathbf{B}_i = \mathbf{A}\mathbf{R}_i)$, where the matrix $\mathbf{A}$ is common to every signer. To sign a message $\mathbf{m}_i$, user i samples a short preimage $\mathbf{v}_i = [\mathbf{v}_{1,i}^T | \mathbf{v}_{2,i}^T]^T$ of $\mathcal{H}(\mathbf{m}_i)$ using our new method, where $\mathcal{H}$ is modeled as a random oracle. At this stage, it is tempting to simply add the first components $\mathbf{v}_{1,i}$ of each signature and concatenate the (very short) second ones $\mathbf{v}_{2,i}$. This would be correct, but the resulting scheme is completely insecure as we will explain. We then resort to a technique generally used to circumvent rogue-key attacks to ensure security, but with some necessary tweaks.

Concretely, to aggregate the $\mathbf{v}_i$, one first obtains small random weights e_i and computes $\mathsf{sig}_{\mathsf{agg}} = (\mathbf{v}_1 = \sum_i e_i \mathbf{v}_{1,i}, (\mathbf{v}_{2,i})_i)$. While this technique seems classical, we note that it is not as straightforward to generate suitable e_i as one might think at first glance. Indeed, generating e_i as the output of a single hash function does not seem sufficient to prove security, even in the random oracle model. This problem, which does not arise in classical cyclic groups, was already faced by the authors of [BR21] who circumvented it by weakening the security model. We show that we can avoid this by resorting to two random oracles $\mathcal{H}_f, \mathcal{H}_e$ to generate the weights e_i so as to deal with the peculiarities of the forking lemma. Concretely, we first compute $f = \mathcal{H}_f(\{\mathbf{B}_j, \mathbf{v}_{2,j}, \mathbf{m}_j\}_{1 \leq j \leq N})$,

and then $e_i = \mathcal{H}_e(f, i) \in \mathcal{C}$ for all i, where $\mathcal{C}$ is the set of ternary polynomials with fixed Hamming weight. To verify, one can then recompute the weights e_i and check that $\mathbf{A}\mathbf{v}_1 + \sum_i e_i(\mathbf{G} - \mathbf{B}_i)\mathbf{v}_{2,i} = \sum_i e_i \mathcal{H}(\mathbf{m}_i)$. We thus manage to prove security according to the conventional model for aggregate signatures at the cost of only one additional call to a hash function.

We only achieve partial aggregation because of the fact that $\mathbf{v}_{2,i}$ faces the matrix $\mathbf{B}_i$ which differs for every user. As a result, we need to transmit all the individual $\mathbf{v}_{2,i}$, thus yielding a size linear in N. However, because our new preimage sampling algorithm minimizes the size of the $\mathbf{v}_{2,i}$'s, it amortizes this linear dependency, enough to have relevant compression compared to the naive concatenation. In particular, we obtain aggregate signatures that are 5% to 16% smaller than the concatenation for N ranging from 10 to 1200 which is a range coherent with real-life applications, such as certificate chains, blockchains or batch software updates for example.

1.2 Organization

We start by recalling some notations and standard notions in Section 2. Then, we provide our new results on preimage sampling in Section 3, and discuss performances in Section 4. We then apply them to design of our lattice-based aggregate signature in Section 5.

2 Preliminaries

In this paper, for two integers $a \leq b$, we define $[a, b] = \{k \in \mathbb{Z} : a \leq k \leq b\}$. When $a = 1$, we simply use $[b]$ instead of $[1, b]$. Further, q is a positive integer, and we define $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. We may identify the latter with the set of representatives $(-q/2, q/2] \cap \mathbb{Z}$. Vectors are written in bold lowercase letters $\mathbf{a}$ and matrices in bold uppercase letters $\mathbf{A}$. The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^T$. The identity matrix of dimension d is denoted by $\mathbf{I}_d$. We use $\|\cdot\|_p$ to denote the ℓ_p norm of $\mathbb{R}^d$, i.e., $\|\mathbf{a}\|_p = (\sum_{i\in[d]} |a_i|^p)^{1/p}$ for any positive integer p, and $\|\mathbf{a}\|_\infty = \max_{i\in[d]} |a_i|$. We also define the spectral norm of a matrix $\mathbf{A}$ by $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq\mathbf{0}} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$. For a finite set S, we define $|S|$ to be its cardinality, and $U(S)$ to be the uniform probability distribution over S. We use $x \hookleftarrow P$ to describe the action of sampling $x \in S$ according to the probability distribution P. In contrast, we use $x \sim P$ to mean that the random variable x follows P. The *statistical distance* between two discrete distributions P, Q over a countable set S is defined as $\Delta(P, Q) = \frac{1}{2} \sum_{x\in S} |P(x) - Q(x)|$. Later, $\mathscr{D}_s, \mathscr{D}_t$ denote arbitrary distributions called source and target distributions respectively.

2.1 Lattices

A full-rank *lattice* $\mathcal{L}$ of rank d is a discrete additive subgroup of $\mathbb{R}^d$. The *dual lattice* of $\mathcal{L}$ is defined by $\mathcal{L}^* = \{\mathbf{x} \in \mathrm{Span}_{\mathbb{R}}(\mathcal{L}) : \forall \mathbf{y} \in \mathcal{L}, \mathbf{x}^T\mathbf{y} \in \mathbb{Z}\}$. We call $\mathsf{Vol}\ \mathcal{L}$ the *volume* of a lattice $\mathcal{L}$. For d, m, q positive integers, we consider the family of

lattices $\{\mathcal{L}_q^\perp(\mathbf{A}); \mathbf{A} \in \mathbb{Z}_q^{d\times m}\}$, where $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\mathbb{Z}\}$. For any $\mathbf{A} \in \mathbb{Z}_q^{d\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^d$, we define $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod q\mathbb{Z}\}$ which is a coset of $\mathcal{L}_q^\perp(\mathbf{A})$.

2.2 Probabilities

For $\mathbf{x}$ a discrete random variable over a set S, we define its min-entropy as $H_\infty(\mathbf{x}) = -\log_2(\max_{\mathbf{x}'\in S} \mathbb{P}_{\mathbf{x}}[\mathbf{x} = \mathbf{x}'])$. We give here the leftover hash lemma from [DORS08] for which we write to match our context and notations.

Lemma 2.1 (Adapted from [DORS08]). *Let d, m_1, q be positive integers such that q is an odd prime. For $\mathbf{A} \sim U(\mathbb{Z}_q^{d\times m_1})$, $\mathbf{x}$ a random variable over $\mathbb{Z}^{m_1}$, and $\mathbf{u} \sim U(\mathbb{Z}_q^d)$, it holds that $\Delta((\mathbf{A}, \mathbf{A}\mathbf{x}), (\mathbf{A}, \mathbf{u})) \leq \frac{1}{2}\sqrt{q^d 2^{-H_\infty(\mathbf{x})}}$. In particular, whenever $H_\infty(\mathbf{x}) \geq d\log_2 q + \omega(\log_2 \lambda)$, the statistical distance is negligible in λ.*

For a center $\mathbf{c} \in \mathbb{R}^d$ and positive definite $\mathbf{S} \in \mathbb{R}^{d\times d}$, we define the Gaussian function $\rho_{\sqrt{\mathbf{S}},\mathbf{c}} : \mathbf{x} \in \mathbb{R}^d \mapsto \exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c}))$. For a countable set $A \subseteq \mathbb{R}^d$, we define the *discrete Gaussian distribution* $\mathcal{D}_{A,\sqrt{\mathbf{S}},\mathbf{c}}$ of support A, covariance $\mathbf{S}$ and center $\mathbf{c}$ by its density $\mathcal{D}_{A,\sqrt{\mathbf{S}},\mathbf{c}} : \mathbf{x} \in A \mapsto \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x})/\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(A)$, where $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(A) = \sum_{\mathbf{x}\in A} \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x})$. When $\mathbf{c} = \mathbf{0}$, we omit it from the notations. When $\mathbf{S} = s^2\mathbf{I}_d$, we use s as subscript instead of $\sqrt{\mathbf{S}}$. As coined by Micciancio and Regev [MR07], we define the *smoothing parameter* of a lattice $\mathcal{L}$, parameterized by $\varepsilon > 0$, by $\eta_\varepsilon(\mathcal{L}) = \inf\{s > 0 : \rho_{1/s}(\mathcal{L}^*) = 1+\varepsilon\}$. We recall the following result stating that $\mathcal{D}_{\mathcal{L},s,\mathbf{c}}$ carries a good amount of entropy when s is sufficiently large. A similar result is given in [PR06, Lem. 2.10], but we give a tighter bound directly resulting from Poisson's summation formula. We give the proof for completeness.

Lemma 2.2. *Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d. For any $\varepsilon > 0$, $s \geq \eta_\varepsilon(\mathcal{L})$, and $\mathbf{c} \in \mathbb{R}^d$, it holds that $H_\infty(\mathcal{D}_{\mathcal{L},s,\mathbf{c}}) \geq d\log_2 s - \log_2(\mathsf{Vol}\ \mathcal{L}) + \log_2(1-\varepsilon)$. In particular, when $\mathcal{L} = \mathbb{Z}^d$ and $\varepsilon \leq 1/2$, it yields $H_\infty(\mathcal{D}_{\mathbb{Z}^d,s}) \geq d\log_2 s - 1$.*

Proof. Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d, $\varepsilon > 0$, $s \geq \eta_\varepsilon(\mathcal{L})$ and $\mathbf{c} \in \mathbb{R}^d$. We look at $\rho_{s,\mathbf{c}}(\mathcal{L})$. By the Poisson summation formula, it holds that

$$\rho_{s,\mathbf{c}}(\mathcal{L}) = s^d(\mathsf{Vol}\ \mathcal{L})^{-1} \sum_{\mathbf{x}\in\mathcal{L}^*} e^{-i\cdot 2\pi\mathbf{x}^T\mathbf{c}}\rho_{1/s}(\mathbf{x}).$$

Yet, it holds that $\left|\sum_{\mathbf{x}\in\mathcal{L}^*} e^{-i\cdot 2\pi\mathbf{x}^T\mathbf{c}}\rho_{1/s}(\mathbf{x}) - 1\right| \leq \rho_{1/s}(\mathcal{L}^* \setminus \{\mathbf{0}\}) \leq \varepsilon$, as $s \geq \eta_\varepsilon(\mathcal{L})$. Since the sum is a positive real, it yields that the latter is bounded below by $1-\varepsilon$. Thence,

$$\rho_{s,\mathbf{c}}(\mathcal{L}) \geq s^d(\mathsf{Vol}\ \mathcal{L})^{-1}(1-\varepsilon).$$

Since $\rho_{s,\mathbf{c}}(\mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{L}$, we have that $H_\infty(\mathcal{D}_{\mathcal{L},s,\mathbf{c}}) \geq \log_2 \rho_{s,\mathbf{c}}(\mathcal{L})$, which gives the desired inequality. When $\mathcal{L} = \mathbb{Z}^d$ and $\varepsilon \leq 1/2$, we have $\mathsf{Vol}\ \mathcal{L} = 1$ and $\log_2(1-\varepsilon) \geq -1$, which yields the claim. □

We also give the standard tail bounds for the discrete Gaussian distribution from [Ban93,Lyu12]. Notice that when $\mathbf{c} = \mathbf{0}$, the usual requirement $s \geq \eta_\varepsilon(\mathcal{L})$ in the following results is not needed.

Lemma 2.3. *Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d. Let $s > 0$ and $\mathbf{v} \in \mathbb{R}^d$. Then, for all $t > 0$, it holds that*

1. $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{L},s}}\left[\|\mathbf{x}\|_2 > s\sqrt{d}\right] < 2^{-2d}$, *[Ban93, Lem. 1.5]*
2. $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{L},s}}\left[|\langle \mathbf{x},\mathbf{v}\rangle| > st\|\mathbf{v}\|_2\right] \leq 2e^{-\pi t^2}$. *[Lyu12, Lem 4.3]*

Based on probabilistic bounds on the spectral norm of sub-Gaussian matrices and on tail bounds of sub-exponential random vectors, we have the following result, proven in e.g. [JRS22].

Lemma 2.4 (Adapted from [JRS22]). *Let m_1, m_2, η be three positive integers and $x, t > 0$. We assume that $m_1 > x \cdot 10/\log_2 e$. Let $\mathbf{x} \in \mathbb{Z}^{m_2}$ such that $\|\mathbf{x}\|_\infty \leq \eta$. We have*

$$\mathbb{P}_{\mathbf{R}\hookleftarrow U([-1,1]^{m_1\times m_2})}[\|\mathbf{R}\mathbf{x}\|_2 \geq \eta\sqrt{m_2}\min(2\sqrt{m_1}, \sqrt{m_1}+\sqrt{m_2}+t)] \leq 2^{-x}+2e^{-\pi t^2},$$

Finally, we give the rejection sampling results from [Lyu12, Thm. 4.6, Lem. 4.7], which were slightly adapted in [JRS22].

Lemma 2.5 (Adapted from [Lyu12, Thm. 4.6, Lem. 4.7]). *Let d be a positive integer, and V, X two countable set of $\mathbb{R}^d$. Let T be a positive real, and we define $V_T = \{\mathbf{v} \in V : \|\mathbf{v}\|_2 \leq T\}$. Let h be a probability distributions on V such that $\mathbb{P}_{\mathbf{v}\sim h}[\mathbf{v} \notin V_T] \leq \varepsilon'$ for some $\varepsilon' \geq 0$. Let $\mathscr{D}_t$ be a probability distribution on X, and $(\mathscr{D}_s^{(\mathbf{v})})_{\mathbf{v}\in V}$ a family of probability distributions on X such that*

$$\exists M > 0, \forall \mathbf{v} \in V_T, \mathbb{P}_{\mathbf{x}\sim\mathscr{D}_t}[M \cdot \mathscr{D}_s^{(\mathbf{v})}(\mathbf{x}) \geq \mathscr{D}_t(\mathbf{x})] \geq 1 - \varepsilon'',$$

for some $\varepsilon'' \geq 0$. We then define two distributions

$\mathcal{P}_1$**:** *Sample $\mathbf{v} \hookleftarrow h$, $\mathbf{x} \hookleftarrow \mathscr{D}_s^{(\mathbf{v})}$. Output $(\mathbf{v}, \mathbf{x})$ with probability $\min(1, \frac{\mathscr{D}_t(\mathbf{x})}{M\mathscr{D}_s^{(\mathbf{v})}(\mathbf{x})})$.*
$\mathcal{P}_2$**:** *Sample $\mathbf{v} \hookleftarrow h$, $\mathbf{x} \hookleftarrow \mathscr{D}_t$. Output $(\mathbf{v}, \mathbf{x})$ with probability $1/M$.*

The outputs of $\mathcal{P}_1$ and $\mathcal{P}_2$ conditioned on not aborting are within statistical distance $\frac{\varepsilon''}{M} + \frac{\varepsilon'(M+1)}{2M}$.

2.3 Algebraic Number Theory

We start by giving the necessary background in algebraic number theory. Our scheme is instantiated over a power-of-two cyclotomic ring. We take n a power of two and let ζ be a primitive $2n$-th root of unity. We define by $K = \mathbb{Q}(\zeta) \cong \mathbb{Q}[X]/\langle X^n + 1\rangle$ the $2n$-th cyclotomic field, and by $R = \mathbb{Z}[\zeta] \cong \mathbb{Z}[X]/\langle X^n + 1\rangle$ its ring of integers. We also define $R_q = R/qR \cong \mathbb{Z}_q[X]/\langle X^n + 1\rangle$ for any modulus $q \geq 2$.

Embeddings. Field and ring elements can be naturally embedded into $\mathbb{R}^n$ by their coefficient vector when seen as polynomials in ζ or X. We call τ the coefficient embedding of R, i.e., for all $r = \sum_{i\in[0,n-1]} r_i\zeta^i \in R$, $\tau(r) = [r_0|\dots|r_{n-1}]^T$. One can extend τ to vectors of R^d by concatenating the coefficient embeddings of each vector entry. Using τ, for a matrix $\mathbf{A} \in R_q^{d\times m}$, $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in R^m : \mathbf{Ax} = \mathbf{0} \bmod qR\}$ embeds into a lattice of $\mathbb{R}^{nm}$ called module lattice. For an integer η, we define $S_\eta = \tau^{-1}([-\eta,\eta]^n)$ and $T_\eta = \tau^{-1}([0,\eta-1]^n)$. We also define the usual norms $\|\cdot\|_p$ over R by $\|r\|_p := \|\tau(r)\|_p$.

Another way to embed R (or its fraction field K) is the canonical embedding, which we denote by σ. More precisely, K has exactly n field homomorphism $\sigma_1,\dots,\sigma_n$ from K to $\mathbb{C}$ which are characterized by the fact that each σ_i maps ζ to one of the distinct roots α_i of X^n+1. The canonical embedding of K is then the ring homomorphism $\sigma(\cdot) = [\sigma_1(\cdot)|\dots|\sigma_n(\cdot)]^T$ from K to $\mathbb{C}^n$ (with entry-wise addition and multiplication of vectors). The canonical and coefficient embeddings are linked linearly by the Vandermonde matrix $\mathbf{V}$ of the α_i, i.e., $\sigma(\cdot) = \mathbf{V}\tau(\cdot)$ with $\mathbf{V} = [\alpha_i^{j-1}]_{i,j\in[n]}$. We note that since the α_i are the n-th roots of -1, they all have magnitude 1. Additionally, in this power-of-two cyclotomic ring, $\mathbf{P} = \mathbf{V}/\sqrt{n}$ is a unitary matrix, i.e., $\mathbf{P}^H\mathbf{P} = \mathbf{I}_n$.

Multiplication Matrices. For all $r, s \in R$, $\tau(rs) = M_\tau(r)\tau(s)$, where $M_\tau(r)$ is the multiplication matrix of $\mathbb{R}^{n\times n}$ associated to r with respect to τ, which in this ring equals

$$M_\tau(r) = \begin{bmatrix} r_0 & -r_{n-1} & \dots & -r_1 \\ r_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -r_{n-1} \\ r_{n-1} & \dots & r_1 & r_0 \end{bmatrix}$$

In the canonical embedding, we obtain $\sigma(rs) = M_\sigma(r)\sigma(s)$ where $M_\sigma(r) = \mathrm{diag}(\sigma_1(r),\dots,\sigma_n(r)) \in \mathbb{C}^{n\times n}$. The link between σ and τ implies that $M_\tau(\cdot) = \mathbf{V}^{-1}M_\sigma(\cdot)\mathbf{V} = \mathbf{P}^H M_\sigma(\cdot)\mathbf{P}$.

Gaussians. We define the discrete Gaussian distribution over R by $\tau^{-1}(\mathcal{D}_{\tau(R),s})$, which we denote by $\mathcal{D}_{R,s}$. Since $\tau(R) = \mathbb{Z}^n$, the distribution corresponds to sampling an integer vector according to $\mathcal{D}_{\mathbb{Z}^n,s}$ which thus defines a ring element via τ^{-1}. One result which we need for our aggregate signature is that the weighted sum of discrete Gaussian vectors over R is also a discrete Gaussian. The result is due to [MP13, Thm. 3.3] which was adapted to the ring setting in [BTT22, Lem. 2.7]. The latter is however formulated with constraints in the canonical embedding σ. We adapt the lemma statement to use the coefficient embedding instead.

Lemma 2.6 (Adapted from [BTT22, Lem. 2.7]). *Let d and N be positive integers. Let $e_1,\dots,e_N$ be arbitrary elements of R, and $s > 0$ such that $s \geq \sqrt{2}\eta_\varepsilon(\mathbb{Z}^{nd}) \cdot \max_{j\in[N]}\|M_\tau(e_i)\|_2$ for a negligible ε. Then it holds that*

$$\Delta\left(\sum_{i\in[N]} e_i\mathcal{D}_{R^d,s}, \mathcal{D}_{\mathcal{L}_e,\sqrt{\mathbf{S}}}\right) \leq \mathsf{negl}(\lambda),$$

where $\mathbf{S} = \mathbf{I}_d \otimes \sum_{i\in[N]} s^2 M_\tau(e_i) M_\tau(e_i)^T$ *and* $\mathcal{L}_e = \sum_{i\in[N]} e_i R^d$ *is a submodule of* R^d.

2.4 Hardness Assumptions

The security of our aggregate signature scheme is based on the *Module Short Integer Solution* (M-SIS) and *Module Learning With Errors* (M-LWE) problems [LS15], which we now recall. We consider both problems in their Hermite Normal Form, i.e., we specify the identity in the M-SIS matrix, and we use the same distribution for the M-LWE secret and error.

Definition 2.1 (M-SIS). *Let* n *be a power-of-two and* $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$. *Let* d, m, q *be positive integers and* $\beta > 0$ *with* $m > d$. *The Module Short Integer Solution problem in Hermite Normal Form* $\text{M-SIS}_{n,d,m,q,\beta}$ *asks to find* $\mathbf{x} \in \mathcal{L}_q^\perp([\mathbf{I}_d|\mathbf{A}']) \setminus \{\mathbf{0}\}$ *such that* $\|\mathbf{x}\|_2 \leq \beta$, *given* $\mathbf{A}' \hookleftarrow U(R_q^{d\times m-d})$.

The advantage of a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ against $\text{M-SIS}_{n,d,m,q,\beta}$ is defined by

$$\text{Adv}_{\text{M-SIS}}[\mathcal{A}] = \mathbb{P}\left[[\mathbf{I}_d|\mathbf{A}']\mathbf{x} = \mathbf{0} \bmod qR \wedge 0 < \|\mathbf{x}\|_2 \leq \beta : \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}')\right],$$

where the probability is over the randomness of $\mathbf{A}'$ and the random coins of $\mathcal{A}$. When the parameters are clear from the context, we define the hardness bound as $\varepsilon_{\text{M-SIS}} = \sup_{\mathcal{A}\ \text{PPT}} \text{Adv}_{\text{M-SIS}}[\mathcal{A}]$. We now present the M-LWE problem in its variant with multiple secrets which we use throughout the paper.

Definition 2.2 (M-LWE). *Let* n *be a power-of-two and* $R = \mathbb{Z}[X]/\langle X^n + 1\rangle$. *Let* d, m, k, q *be positive integers and* $\mathscr{D}_r$ *a distribution on* R. *The Module Learning With Errors problem* $\text{HNF-M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$ *asks to distinguish between the following distributions: (1)* $(\mathbf{A}', [\mathbf{I}_m|\mathbf{A}']\mathbf{R} \bmod qR)$, *where* $\mathbf{A}' \sim U(R_q^{m\times d})$ *and* $\mathbf{R} \sim \mathcal{D}_r^{d+m\times k}$, *and (2)* $(\mathbf{A}', \mathbf{B})$, *where* $\mathbf{A}' \sim U(R_q^{m\times d})$ *and* $\mathbf{B} \sim U(R_q^{m\times k})$.

The advantage of a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ against $\text{M-LWE}^k_{n,d,m,q,\mathcal{D}_r}$ is defined by

$$\text{Adv}_{\text{M-LWE}}[\mathcal{A}] = |\mathbb{P}\left[\mathcal{A}(\mathbf{A}', [\mathbf{I}_m|\mathbf{A}']\mathbf{R}) = 1\right] - \mathbb{P}\left[\mathcal{A}(\mathbf{A}', \mathbf{B}) = 1\right]|,$$

When the parameters are clear from the context, we define the hardness bound as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{A}\ \text{PPT}} \text{Adv}_{\text{M-LWE}}[\mathcal{A}]$. When $n = 1$, we use the notation $\text{LWE}^k_{d,m,q,\mathcal{D}_r}$ to denote the same problem over $R = \mathbb{Z}$. Additionally, a standard hybrid argument shows that $\text{M-LWE}^k_{n,d,m,q,\mathcal{D}_r}$ is at least as hard as $\text{M-LWE}^1_{n,d,m,q,\mathcal{D}_r}$ at the expense of a loss factor k in the reduction.

2.5 General Forking Lemma

We give here the general forking lemma from Bellare and Neven [BN06] in Lemma 2.7 and the forking algorithm $\mathcal{F}_\mathcal{B}$ in Algorithm 2.1. We later need this result to prove the security of our aggregate signature scheme in Section 5.3.

Lemma 2.7 ([BN06, Lem. 1]). *Let Q_e be a positive integer and $\mathcal{C}$ a set of size at least 2. Let $\mathcal{B}$ be a randomized algorithm that on input $x, h_1, \ldots, h_{Q_e}$ returns a pair consisting of an integer in $\{0, \ldots, Q_e\}$ and a second element referred to as a side output. Let IG be a randomized algorithm that we call input generator. We define the accepting probability as*

$$\mathsf{acc} = \mathbb{P}[j \geq 1 : x \leftarrow \mathsf{IG}; h_1, \ldots, h_{Q_e} \hookleftarrow U(\mathcal{C}); (j, \mathsf{out}) \leftarrow \mathcal{B}(x, h_1, \ldots, h_{Q_e})].$$

The forking algorithm $\mathcal{F}_{\mathcal{B}}$ associated to $\mathcal{B}$ takes as input x and is described in Algorithm 2.1. We define the probability

$$\mathsf{frk} = \mathbb{P}[b = 1 : x \leftarrow \mathsf{IG}; (b, \mathsf{out}, \mathsf{out}') \leftarrow \mathcal{F}_{\mathcal{B}}(x)].$$

Then, it holds that $\mathsf{acc} \leq Q_e/|\mathcal{C}| + \sqrt{Q_e \cdot \mathsf{frk}}$

Algorithm 2.1: Forking $\mathcal{F}_{\mathcal{B}}$

On input x, proceed as follows.
1. Pick random coins ρ for $\mathcal{B}$
2. $h_1, \ldots, h_{Q_e} \hookleftarrow U(\mathcal{C})$
3. $(j, \mathsf{out}) \leftarrow \mathcal{B}(x, h_1, \ldots, h_{Q_e}; \rho)$
4. **if** $j = 0$, **return** $(0, \perp, \perp)$
5. $h'_j, \ldots, h'_{Q_e} \hookleftarrow U(\mathcal{C})$
6. $(j', \mathsf{out}') \leftarrow \mathcal{B}(x, h_1, \ldots, h_{j-1}, h'_j, \ldots, h'_{Q_e}; \rho)$
7. **if** $(j = j') \wedge (h_j \neq h'_j)$, **return** $(1, \mathsf{out}, \mathsf{out}')$
8. **else return** $(0, \perp, \perp)$.

3 Revisiting Trapdoor Sampling

We here focus on the trapdoor preimage sampling procedure proposed by Micciancio and Peikert [MP12]. In Section 3.2, we show that a finer analysis of the perturbation sampling step allows one to generate preimages that are 20% smaller without adding any requirement and at absolutely no cost on the security. As a result, this can be used as a drop-in replacement in every scheme using trapdoors from [MP12] and preimage sampling. This relies on the observation that preimages $\mathbf{v}$ are in two parts $\mathbf{v}_1, \mathbf{v}_2$ which have asymmetric roles but are treated symmetrically in [MP12]. By slightly breaking this symmetry, we are able to significantly reduce the size of $\mathbf{v}_2$, which leads to the gain mentioned above.

In a second step, we show in Section 3.3 that we can leverage further this asymmetry using the preimage sampler of Lyubashevsky and Wichs [LW15]. The analysis of said sampler provided by the authors however places very restrictive constraints on the parameters that make it less efficient than the best instantiation of the original sampler of [MP12]. We provide an improved analysis of the sampler which gets rid of those parameter constraints, leading to a performance improvement of around 30% over the original sampler from [MP12]. The resulting sampler is then more efficient than our first optimization mentioned above but places moderate constraints on the applications, which are easily met in

practice as we discuss. The special features of the resulting preimages could also have other consequences on some specific primitives. As an example, Section 5 presents the first lattice-based aggregate signature scheme that supports public aggregation with relevant compression.

3.1 Micciancio-Peikert Preimage Sampling

The notion of trapdoors introduced by Micciancio and Peikert [MP12] (which we later abbreviate MP trapdoors) is very versatile and has been extensively used in cryptographic constructions, including many advanced lattice-based primitives. In particular, it yields the ability to naturally design tag-based constructions, a property leveraged in a number of works such as group signatures [dPLS18,LNPS21] or signature with efficient protocols [JRS22]. This new notion of trapdoors also allows for more efficient preimage sampling due to the specific form of the trapdoor function. More precisely, they generate matrices $\mathbf{A_H}$ of the form

$$\mathbf{A_H} = [\mathbf{A}|\mathbf{HG} - \mathbf{AR}] \bmod q\mathbb{Z} \in \mathbb{Z}_q^{d\times(m_1+m_2)},$$

where $\mathbf{H} \in \mathbb{Z}_q^{d\times d}$ is an invertible tag matrix, $\mathbf{G} \in \mathbb{Z}^{d\times m_2}$ a primitive gadget matrix, and $\mathbf{R} \in \mathbb{Z}^{m_1\times m_2}$ a short matrix corresponding to the trapdoor. The advantage of such a construction is that the same trapdoor information $\mathbf{R}$ can be used for all tags $\mathbf{H}$. The gadget $\mathbf{G}$ is chosen so that it is easy to compute short preimages, and therefore, it becomes easy to compute preimages of $\mathbf{A_H}$ with the knowledge of $\mathbf{R}$. In what follows, we consider the gadget matrix of [MP12] in base $b \geq 2$, i.e., $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\dots|b^{\lceil \log_b q\rceil -1}] \in \mathbb{Z}^{d\times m_2}$ where $m_2 = d\lceil \log_b q\rceil$.

The sampling algorithm relies on the link between such matrices $\mathbf{A_H}$ and the gadget matrix $\mathbf{G}$, that is

$$\mathbf{A_H}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_{m_2}\end{bmatrix} = \mathbf{HG} \bmod q\mathbb{Z}.$$

Thence, if $\mathbf{z}$ is a short vector in $\mathcal{L}_q^{\mathbf{u}}(\mathbf{HG})$, then $\mathbf{v} = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$ is a short vector in $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_H})$, i.e., verifying $\mathbf{A_H v} = \mathbf{u} \bmod q\mathbb{Z}$, that is $\mathbf{v}$ is a preimage of $\mathbf{u}$ by $\mathbf{A_H}$. Unfortunately, $\mathbf{v}$ leaks information about the trapdoor $\mathbf{R}$ which is undesirable in cryptographic applications as $\mathbf{R}$ usually corresponds to the long-term secret key. To circumvent this issue, the authors use the Gaussian convolution theorem [Pei10, Thm. 3.1] to perturb $\mathbf{v}$ in order to make the final samples independent of $\mathbf{R}$. In more details, they sample a (highly) non-spherical Gaussian perturbation $\mathbf{p} = [\mathbf{p}_1^T|\mathbf{p}_2^T]^T \sim \mathcal{D}_{\mathbb{Z}^{m_1+m_2},\sqrt{\mathbf{S}}}$ with

$$\mathbf{S} = s^2\mathbf{I}_{m_1+m_2} - s_{\mathbf{G}}^2\begin{bmatrix}\mathbf{R}\mathbf{R}^T & \mathbf{R}\\ \mathbf{R}^T & \mathbf{I}_{m_2}\end{bmatrix},$$

and then compensate this perturbation by sampling $\mathbf{z} \sim \mathcal{D}_{\mathcal{L}_q^{\mathbf{x}}(\mathbf{G}),s_{\mathbf{G}}}$ with $\mathbf{x} = \mathbf{H}^{-1}(\mathbf{u}-\mathbf{Ap}_1+\mathbf{ARp}_2)-\mathbf{Gp}_2$. The output sample is then $\mathbf{v}' = [(\mathbf{p}_1+\mathbf{Rz})^T|(\mathbf{p}_2+$

$\mathbf{z})^T]^T$. By the convolution theorem, $\mathbf{v}'$ is statistically close to a Gaussian distribution over $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_H})$ with parameter s, which no longer depends on $\mathbf{R}$.

Therefore, from the security standpoint, the approach above perfectly addresses the problem of preimage sampling for cryptographic applications. However, if we reconsider the unperturbed vector $\mathbf{v} = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$, we note that the convolution is now applied to both parts in the same way. This does not seem optimal as the bottom section of $\mathbf{v}$ is independent of $\mathbf{R}$ and as $\mathbf{Rz}$ is always larger than $\mathbf{z}$. Unfortunately, this seems inherent to the approach stated in [Pei10, Sec. 1.3] which only considers covariance matrices of the form $s^2\mathbf{I} - \mathbf{S}_1$ for some covariance matrix $\mathbf{S}_1$. Ideally, we would like to select a perturbation that only affects the top component, typically:

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{0} \end{bmatrix} \sim \mathcal{D}_{\mathbb{Z}^{m_1+m_2}, \sqrt{\mathbf{S}}}, \text{ with } \mathbf{S} = \begin{bmatrix} s^2\mathbf{I}_{m_1} - s_{\mathbf{G}}^2\mathbf{R}\mathbf{R}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

However, when sampling $\mathbf{z}$ and outputting $\mathbf{p} + [\mathbf{R}^T|\mathbf{I}_{m_2}]^T\mathbf{z}$, we end up with a joint probability of covariance

$$\begin{bmatrix} s^2\mathbf{I}_{m_1} - s_{\mathbf{G}}^2\mathbf{R}\mathbf{R}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + s_{\mathbf{G}}^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^T & \mathbf{R} \\ \mathbf{R}^T & \mathbf{I}_{m_2} \end{bmatrix} = \begin{bmatrix} s^2\mathbf{I}_{m_1} & s_{\mathbf{G}}^2\mathbf{R} \\ s_{\mathbf{G}}^2\mathbf{R}^T & s_{\mathbf{G}}^2\mathbf{I}_{m_2} \end{bmatrix},$$

which again leaks information about $\mathbf{R}$. This highlights the fact that in order to rely on the convolution technique, one needs to hide both $\mathbf{Rz}$ and $\mathbf{z}$. Intuitively, the first component $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ can be seen as a Gaussian distribution with a secret center $\mathbf{Rz}$. Looking at its marginal distribution, one could use standard techniques to hide this secret center, namely convolution when $\mathbf{z}$ is Gaussian or noise flooding (based on either the statistical distance or the Rényi divergence)if $\mathbf{z}$ is non-Gaussian. However, giving $\mathbf{v}_2 = \mathbf{z}$ provides side information on this secret center which explains why $\mathbf{z}$ also has to be perturbed for the convolution technique to be meaningful. We therefore need a middle way between this efficient, but insecure, approach and the one from [MP12] that does not seem optimal for the type of asymmetric vectors we have to perturb.

3.2 Finer Analysis of Perturbation Sampling

Our first solution is to break the symmetry between the top and bottom parts in [MP12] by using different parameters s_1 and s_2. More precisely, we sample a perturbation over $\mathbb{Z}^{m_1+m_2}$ of covariance

$$\mathbf{S} = \begin{bmatrix} s_1^2\mathbf{I}_{m_1} & \mathbf{0} \\ \mathbf{0} & s_2^2\mathbf{I}_{m_2} \end{bmatrix} - s_{\mathbf{G}}^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^T & \mathbf{R} \\ \mathbf{R}^T & \mathbf{I}_{m_2} \end{bmatrix},$$

where s_2 will hopefully be much smaller than s_1 because $\mathbf{z}$ has to be perturbed by a smaller amount than $\mathbf{Rz}$. The natural question is then to determine how small it can be. At this stage we note that the reasoning in [Pei10, Sec. 1.3] is of no help here as $\mathbf{S}$ is no longer of the form $s^2\mathbf{I} - \mathbf{S}_1$. We therefore need a new result tailored to our need so as to derive bounds on s_1 and s_2. More specifically, to continue using the convolution theorem in [Pei10], we need $\mathbf{S}$ to be positive definite, leading to the following lemma.

Lemma 3.1. *Let m, ℓ be positive integers, $\mathbf{R} \in \mathbb{R}^{m\times\ell}$, and α, β, γ positive reals. If $\alpha > \sqrt{2}\cdot\gamma\|\mathbf{R}\|_2$ and $\beta > \sqrt{2}\gamma$, then the matrix*

$$\mathbf{S} = \begin{bmatrix} \alpha^2\mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \beta^2\mathbf{I}_\ell \end{bmatrix} - \gamma^2 \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_\ell \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & \mathbf{I}_\ell \end{bmatrix}$$

is positive definite.

Proof. We first consider the singular value decomposition of $\mathbf{R}$ as $\mathbf{R} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, with $\mathbf{U} \in \mathbb{R}^{m\times m}$ unitary, $\mathbf{V} \in \mathbb{R}^{\ell\times\ell}$ unitary, and $\mathbf{S} \in \mathbb{R}^{m\times\ell}$ a diagonal matrix with non-negative entries in decreasing order. Using the fact that $\mathbf{U}, \mathbf{V}$ are unitary, we have

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \left(\begin{bmatrix} \alpha^2\mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \beta^2\mathbf{I}_\ell \end{bmatrix} - \gamma^2 \begin{bmatrix} \mathbf{S}\mathbf{S}^T & \mathbf{S} \\ \mathbf{S}^T & \mathbf{I}_\ell \end{bmatrix} \right) \begin{bmatrix} \mathbf{U}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^T \end{bmatrix}$$

Let $\mathbf{x} = [\mathbf{x}_1^T|\mathbf{x}_2^T]^T \in \mathbb{R}^{m+\ell} \setminus \{\mathbf{0}\}$ with $\mathbf{x}_1 \in \mathbb{R}^m$ and $\mathbf{x}_2 \in \mathbb{R}^\ell$. We then define $\mathbf{y}_1 = \mathbf{U}^T\mathbf{x}_1$, $\mathbf{y}_2 = \mathbf{V}^T\mathbf{x}_2$ and $\mathbf{y} = [\mathbf{y}_1^T|\mathbf{y}_2^T]^T \neq \mathbf{0}$. Now assume that $m \geq \ell$. We thus have $\mathbf{S} = [\mathbf{D}|\mathbf{0}_{m-\ell\times\ell}]^T$ with $\mathbf{D} = \mathrm{diag}(s_1, \ldots, s_\ell) \in \mathbb{R}^{\ell\times\ell}$. Hence,

$$\begin{aligned}
\mathbf{x}^T\mathbf{\Sigma}\mathbf{x} &= \alpha^2 \sum_{i=1}^{m} y_{1,i}^2 + \beta^2 \sum_{i=1}^{\ell} y_{2,i}^2 - \gamma^2 \sum_{i=1}^{\ell} \left(s_i y_{1,i} + y_{2,i}\right)^2 \\
&\geq \alpha^2 \sum_{i=1}^{m} y_{1,i}^2 + \beta^2 \sum_{i=1}^{\ell} y_{2,i}^2 - \gamma^2 \sum_{i=1}^{\ell} 2\left(s_i^2 y_{1,i}^2 + y_{2,i}^2\right) \\
&= \sum_{i=1}^{\ell} \left((\alpha^2 - 2\gamma^2 s_i^2) y_{1,i}^2 + (\beta^2 - 2\gamma^2) y_{2,i}^2\right) + \sum_{i=\ell+1}^{m} \alpha^2 y_{1,i}^2 \\
&> 0,
\end{aligned}$$

because $\alpha^2 > 2\gamma^2\|\mathbf{R}\|_2^2 = 2\gamma^2 \max_{1\leq i\leq \ell} s_i^2$, and $\beta^2 > 2\gamma^2$. Next, assuming $m \leq \ell$, we have $\mathbf{S} = [\mathbf{D}|\mathbf{0}_{m\times\ell-m}]$ with $\mathbf{D} = \mathrm{diag}(s_1, \ldots, s_m) \in \mathbb{R}^{m\times m}$. Similarly, it yields

$$\begin{aligned}
\mathbf{x}^T\mathbf{\Sigma}\mathbf{x} &= \alpha^2 \sum_{i=1}^{m} y_{1,i}^2 + \beta^2 \sum_{i=1}^{\ell} y_{2,i}^2 - \gamma^2 \sum_{i=1}^{m} \left(s_i y_{1,i} + y_{2,i}\right)^2 - \gamma^2 \sum_{i=m+1}^{\ell} y_{2,i}^2 \\
&\geq \alpha^2 \sum_{i=1}^{m} y_{1,i}^2 + \beta^2 \sum_{i=1}^{\ell} y_{2,i}^2 - \gamma^2 \sum_{i=1}^{m} 2\left(s_i^2 y_{1,i}^2 + y_{2,i}^2\right) - \gamma^2 \sum_{i=m+1}^{\ell} y_{2,i}^2 \\
&\geq \alpha^2 \sum_{i=1}^{m} y_{1,i}^2 + \beta^2 \sum_{i=1}^{\ell} y_{2,i}^2 - \gamma^2 \sum_{i=1}^{m} 2\left(s_i^2 y_{1,i}^2 + y_{2,i}^2\right) - \gamma^2 \sum_{i=m+1}^{\ell} 2y_{2,i}^2 \\
&= \sum_{i=1}^{m} (\alpha^2 - 2\gamma^2 s_i^2) y_{1,i}^2 + \sum_{i=1}^{\ell} (\beta^2 - 2\gamma^2) y_{2,i}^2 \\
&> 0,
\end{aligned}$$

as desired. □

In the context of [MP12], we will have to use the previous lemma on the matrices $\mathbf{S} - \mathbf{I}_{m_1+m_2}$ and $\mathbf{S} - 2[\mathbf{R}^T|\mathbf{I}]^T[\mathbf{R}^T|\mathbf{I}]$. As a result, we must take s_1 and s_2 such that $\sqrt{s_1^2 - 1} > \sqrt{2}s_{\mathbf{G}}\|\mathbf{R}\|_2$ and $\sqrt{s_2^2 - 1} > \sqrt{2}s_{\mathbf{G}}$, as well as $s_1 > \sqrt{2(s_{\mathbf{G}}^2 + 2)}\|\mathbf{R}\|_2$ and $s_2 > \sqrt{2(s_{\mathbf{G}}^2 + 2)}$. The latter two conditions subsum the former two. We recall that we also have to consider the randomized rounding factor $r \geq \eta_\varepsilon(\mathbb{Z})$, typically $r \approx 5.4$. We can therefore set $s_1 > r\sqrt{2s_{\mathbf{G}}^2 + 4}\|\mathbf{R}\|_2$ and $s_2 > r\sqrt{2s_{\mathbf{G}}^2 + 4}$ with $s_{\mathbf{G}} \approx \sqrt{b^2 + 1}$, and still inherit from the analysis of [MP12]. This allows us to drastically reduce the size of the bottom part for free, while keeping the size of the top part (almost) the same as before. Additionally, the overall norm of $\mathbf{v}$ is smaller which can result in slightly increased concrete security. For example, in GPV signatures [GPV08], smaller preimages leads to a smaller SIS bound and in turn better security. This modification can thus be used as is in every scheme using MP trapdoor preimage sampling, leading to better performance as illustrated in Section 4.

3.3 A More Flexible Preimage Sampler

Although we improved the quality of the preimage sampling procedure, it is still quite rigid. Namely, it still requires the sampling of a perturbation vector $\mathbf{p}$ from a (highly) non-spherical Gaussian distribution. Such a perturbation sampling is rather costly and represents the most part of the computation time of preimage sampling. The gadget sampling step (sampling $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^{\mathbf{x}}(\mathbf{G}),s_{\mathbf{G}}}$) also requires the sampling of non-spherical Gaussian perturbations when q is not a power of the gadget base b. However, the latter has been analyzed in several works [GM18,ZY22] by identifying structure in the basis of $\mathcal{L}_q^{\perp}(\mathbf{G})$ to enable more efficient sampling over $\mathcal{L}_q^{\perp}(\mathbf{G})$. But for the perturbation $\mathbf{p}$ we consider, we cannot leverage a particular structure of the covariance matrix $\mathbf{S}$ as $\mathbf{R}$ is generated randomly. Another limitation is that this convolution method is seemingly limited to Gaussian distributions, which in turn limits the possible preimage distributions.

3.3.1 Description. To circumvent these shortcomings, Lyubashevsky and Wichs [LW15] proposed another more flexible preimage sampling procedure which only perturbs the top component. The approach from [LW15] can be seen as combining the features of tag-friendly gadget-based preimage sampling with rejection sampling that is extensively used in Fiat-Shamir with Aborts (FSwA) signatures. Let $\mathbf{G}^{-1}(\cdot)$ be the entry-wise base-b decomposition of vectors of $\mathbb{Z}_q^d$, thus resulting in vectors of $[0, b-1]^{m_2}$. The intuition is to sample a perturbation $\mathbf{p}_1 \in \mathbb{Z}^{m_1}$ from a source distribution $\mathscr{D}_s$. Further, instead of using Gaussian $\mathbf{G}$-sampling, we simply use the base-b decomposition and obtain $\mathbf{v}_2 = \mathbf{G}^{-1}(\mathbf{H}^{-1}(\mathbf{u} - \mathbf{A}\mathbf{p}_1))$. Then, we can define $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$ so that the relation $\mathbf{A}_{\mathbf{H}}\mathbf{v} = \mathbf{u}$ is verified, and apply rejection sampling to make $\mathbf{v}_1$ independent of $\mathbf{R}\mathbf{v}_2$ and in turn $\mathbf{R}$. This setting is reminiscent of lattice-based zero-knowledge arguments or Lyubashevsky's signature scheme [Lyu12], where $\mathbf{R}$ is the witness, $\mathbf{p}_1$ is the mask, $\mathbf{A}\mathbf{p}_1$ is a commitment to the mask, $\mathbf{v}_2$ is the challenge, and $\mathbf{v}_1$

is the response to the challenge. We slightly modify the presentation of the sampler from [LW15] by taking the matrix $\mathbf{A}$ in Hermite Normal Form. Concretely, throughout the rest of the paper, $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ for a matrix $\mathbf{A}'$ of dimension $d \times (m_1 - d)$.

Algorithm 3.1: SamplePre($\mathbf{R}; \mathbf{A}', \mathbf{H}, \mathbf{u}, \mathscr{D}_s, \mathscr{D}_t$)

Input (offline phase)**:** Matrix $\mathbf{A}' \in \mathbb{Z}_q^{d\times(m_1-d)}$, Source distribution $\mathscr{D}_s$ over $\mathbb{Z}^{m_1}$.
Input (online phase)**:** Trapdoor $\mathbf{R} \in \mathbb{Z}^{m_1\times m_2}$, Tag $\mathbf{H} \in GL_d(\mathbb{Z}_q)$, Syndrome $\mathbf{u} \in \mathbb{Z}_q^d$, Target distributions $\mathscr{D}_t$ over $\mathbb{Z}^{m_1}$ such that rejection sampling can be performed with respect to $\mathcal{D}_s$.

Offline phase

1. $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$.
2. $\mathbf{w} \leftarrow [\mathbf{I}_d|\mathbf{A}']\mathbf{p}_1 \bmod q\mathbb{Z}$.

Online phase

3. $\mathbf{x} \leftarrow \mathbf{H}^{-1}(\mathbf{u} - \mathbf{w}) \bmod q\mathbb{Z}$. ▷ Syndrome correction
4. $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{x}) \in [0, b-1]^{m_2}$. ▷ Deterministic. $m_2 = d\lceil \log_b q \rceil$
5. $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$.
6. Sample a continuous $u \hookleftarrow U([0,1])$.
7. **if** $u > \min\left(1, \frac{\mathscr{D}_t(\mathbf{v}_1)}{M\cdot\mathscr{D}_s(\mathbf{p}_1)}\right)$ **then** go back to 1.

Output: $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix}$.

3.3.2 Current Limitations. At first glance, the approach from [LW15] seems to fully achieve what we wanted to do in Section 3.2, namely to completely break the symmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$ to further reduce the size of $\mathbf{v}_2$. However, in practice, the choice of parameters and suitable distributions $\mathscr{D}_s, \mathscr{D}_t$ is conditioned by the security requirements coming from the simulation result of [LW15, Thm. 3.1]. Unfortunately, the latter is too restrictive in most cases, which explains why it does not lead to improvements on the preimage size, as we explain below.

In [LW15, Thm. 3.1], it is shown that the output distribution of SamplePre is statistically close to a distribution that does not depend on the trapdoor $\mathbf{R}$ for an arbitrary (potentially adversarial) syndrome $\mathbf{u}$. It means that a preimage $\mathbf{v}$ of $\mathbf{u}$ can be simulated without resorting to the trapdoor $\mathbf{R}$, and thus does not leak information on $\mathbf{R}$. Because they deal with an arbitrary $\mathbf{u}$, nothing can be assumed about its distribution which in turn places strong restrictions on the parameters to compensate. Indeed, in their result, they need to assume that $\mathbf{A}\mathbf{v}_1$ (and $\mathbf{A}\mathbf{p}_1$) is *statistically* close to uniform requiring the parameters to be large in order to use a regularity lemma. This requirement in turn prevents them from using a (much more efficient) computational instantiation of MP trapdoors where $m_1 = 2d$ so that $\mathbf{AR}$ can be argued to be pseudorandom based on LWE rather than the leftover hash lemma. This results in significant performance losses which cancel out the benefits of having smaller $\mathbf{v}_2$.

3.3.3 An Improved Simulation of Preimages. We now explain how to get rid of this requirements. In many situations in cryptography, the syndrome follows a prescribed distribution. For GPV signatures, and also in our aggre-

gate signature scheme that we present in Section 5, the syndrome $\mathbf{u}$ is the hash output $\mathcal{H}(\mathbf{m})$ of the message $\mathbf{m}$ where $\mathcal{H}$ is modeled as a random oracle. This means that the syndrome we expect are uniformly distributed and cannot be controlled by the adversary. Making this assumption on the distribution of $\mathbf{u}$ allows us to remove this constraint on $\mathbf{A}\mathbf{v}_1$ being statistically close to uniform as we can, at a high level, use the randomness of $\mathbf{u}$ to achieve the same conclusions. Removing this constraint avoids the need for a large perturbation (either in norm or dimension) and thus leads to improved performances. We provide our new simulation result in Theorem 3.1, which we instantiate for Gaussian distributions in Corollary 3.1. Because we assume $\mathbf{u}$ to be uniform in the real distribution, we also need to ensure that the simulated syndromes are indeed uniform. More precisely, we prove that the pairs $(\mathbf{v}, \mathbf{u})$, with $\mathbf{u}$ uniform, can be simulated without resorting to the trapdoor $\mathbf{R}$, and that they indeed do not leak information about $\mathbf{R}$.

This trapdoor-independence property of the preimages is necessary for cryptographic applications, e.g., signatures, as an adversary can usually have access to many such preimages (and syndromes) for a single key. To anticipate such uses, we present the simulation of Q preimages. Looking ahead, Q would later denote the maximal number of emitted signatures per key as in the GPV construction [GPV08].

Theorem 3.1. *Let d, q, b, Q be positive integers with q prime. Let $m_1 = 2d$, $k = \lceil \log_b q \rceil$ and $m_2 = dk$. Let $\mathscr{D}_r, \mathscr{D}_s, \mathscr{D}_t$ be three distributions over $\mathbb{Z}$, $\mathbb{Z}^{m_1}$ and $\mathbb{Z}^{m_1}$ respectively. We denote by $\mathscr{D}'_t$ the marginal distribution of the last $m_1 - d = d$ entries of a random vector sampled from $\mathscr{D}_t$. We then define by h the distribution obtained by sampling $\mathbf{R} \hookleftarrow \mathscr{D}_r^{m_1 \times m_2}$ and $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(\mathbb{Z}_q^d))$ and outputting $\mathbf{R}\mathbf{v}_2$. We denote by $V = Supp(h)$. We let T be a positive real and assume $\mathbb{P}_{\mathbf{y}\sim h}[\|\mathbf{y}\|_2 > T] \leq \varepsilon'$ for some $\varepsilon' \geq 0$. We assume there exists $M > 0$ such that for all $\mathbf{y} \in V$, if $\|\mathbf{y}\|_2 \leq T$, then $\mathbb{P}_{\mathbf{v}_1 \sim \mathscr{D}_t}[M(\mathbf{y} + \mathscr{D}_s)(\mathbf{v}_1) \geq \mathcal{D}_t(\mathbf{v}_1)] \geq 1 - \varepsilon''$ for some $\varepsilon'' \geq 0$.*

Let $\mathbf{A}' \sim U(\mathbb{Z}_q^{d \times d})$, $\mathbf{R} \sim \mathscr{D}_r^{m_1 \times m_2}$, $\mathbf{H} \in GL_d(\mathbb{Z}_q)$ and $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}'] \in \mathbb{Z}_q^{d \times m_1}$. We define the following distributions.

$\mathcal{P}_1$

1. $\mathbf{u}_1, \ldots, \mathbf{u}_Q \hookleftarrow U(\mathbb{Z}_q^d)$.
2. *For all* $i \in [Q]$, $\mathbf{v}_i \leftarrow \mathsf{SamplePre}(\mathbf{R}; \mathbf{A}', \mathbf{H}, \mathbf{u}_i, \mathscr{D}_s, \mathscr{D}_t)$.

Output: $((\mathbf{v}_i)_{i\in[Q]}, (\mathbf{u}_i)_{i\in[Q]})$.

$\mathcal{P}_2$

For all $i \in [Q]$

1. $\mathbf{v}_{1,i} \hookleftarrow \mathscr{D}_t$, $\mathbf{v}_{2,i} \hookleftarrow \mathbf{G}^{-1}(U(\mathbb{Z}_q^d))$.
2. $\mathbf{v}_i \leftarrow [\mathbf{v}_{1,i}^T | \mathbf{v}_{2,i}^T]^T$.
3. $\mathbf{u}_i \leftarrow [\mathbf{A}|\mathbf{H}\mathbf{G} - \mathbf{A}\mathbf{R}]\mathbf{v}_i \bmod q\mathbb{Z}$.
4. *With probability* $1 - 1/M$ *go back to 1. for the same i*

Output: $((\mathbf{v}_i)_{i\in[Q]}, (\mathbf{u}_i)_{i\in[Q]})$.

Then, it holds that the advantage of any PPT distinguisher $\mathcal{A}$ between $\mathcal{P}_1$ and $\mathcal{P}_2$ is at most

$$Adv_{\mathcal{P}_1,\mathcal{P}_2}[\mathcal{A}] \leq \varepsilon_{\mathrm{LWE}} + Q\left(2^{-\frac{1}{2}H_\infty(\mathcal{D}'_t)-1} + \frac{\varepsilon''}{M} + \frac{\varepsilon'(M+1)}{2M}\right),$$

where $\varepsilon_{\mathrm{LWE}}$ *is the hardness bound of* $\mathrm{LWE}^{m_2}_{d,d,q,\mathscr{D}_r}$.

Proof. We first look at the first components $\mathbf{v}_i$. When $\mathbf{u}_i \sim U(\mathbb{Z}_q^d)$, then for $\mathbf{p}_{1,i} \sim \mathscr{D}_s$ independent of $\mathbf{u}_i$, it holds that $\mathbf{x}_i = \mathbf{H}^{-1}(\mathbf{u}_i - \mathbf{A}\mathbf{p}_{1,i}) \bmod q\mathbb{Z}$ is also uniformly distributed in $\mathbb{Z}_q^d$. This is due to the fact that $\mathbf{H}^{-1} \in GL_d(\mathbb{Z}_q)$ and thus preserves the uniform distribution. Note that $\mathbf{v}_{2,i}$ is not uniform in $[0, b-1]^{m_2}$ but in $\mathbf{G}^{-1}(\mathbb{Z}_q^d)$ which is not the same unless $q = b^k$. Hence, we have

$$\Delta((\mathbf{v}_{2,i})_{\mathcal{P}_1}, (\mathbf{v}_{2,i})_{\mathcal{P}_2}) = 0. \tag{1}$$

It thus holds that in $\mathcal{P}_1$, $\mathbf{y}_i = \mathbf{R}\mathbf{v}_{2,i}$ is distributed according to h, and $\mathbf{p}_{1,i} + \mathbf{y}_i$ according to $\mathscr{D}_s + \mathbf{y}_i$. By our assumptions on $h, \mathscr{D}_s, \mathscr{D}_t$, the rejection sampling result of Lemma 2.5 yields that

$$\Delta((\mathbf{R}\mathbf{v}_{2,i}, \mathbf{v}_{1,i})_{\mathcal{P}_1}, (\mathbf{R}\mathbf{v}_{2,i}, \mathbf{v}_{1,i})_{\mathcal{P}_2}) \leq \frac{\varepsilon''}{M} + \frac{\varepsilon'(M+1)}{2M},$$

By the data processing inequality of the statistical distance, it holds

$$\Delta((\mathbf{v}_{1,i})_{\mathcal{P}_1}, (\mathbf{v}_{1,i})_{\mathcal{P}_2}) \leq \frac{\varepsilon''}{M} + \frac{\varepsilon'(M+1)}{2M}. \tag{2}$$

Now let us look at the second components $\mathbf{u}_i$. Let $\mathcal{A}'$ be a distinguisher between $((\mathbf{u}_i)_{\mathcal{P}_1})_i$ and $([\mathbf{A}|\mathbf{H}\mathbf{G} - \mathbf{A}\mathbf{R}](\mathbf{v}_i)_{\mathcal{P}_2})_i$ with advantage δ. We use it to construct a distinguisher $\mathcal{B}$ for $\mathrm{LWE}^{m_2}_{d,d,q,\mathscr{D}_r}$. $\mathcal{B}$ takes as input $(\mathbf{A}', \mathbf{B}) \in \mathbb{Z}_q^{d\times d} \times \mathbb{Z}_q^{d\times m_2}$ with $\mathbf{A}' \hookleftarrow U(\mathbb{Z}_q^{d\times d})$. The distinguisher then samples the $\mathbf{v}_i$ as in $\mathcal{P}_2$ and set $\mathbf{u}_i = [\mathbf{I}_d|\mathbf{A}'|\mathbf{H}\mathbf{G} - \mathbf{B}]\mathbf{v}_i \bmod q\mathbb{Z}$. It then sends $(\mathbf{u}_i)_i$ to $\mathcal{A}'$. If $\mathbf{B} = [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod q\mathbb{Z}$ (LWE case), then the input to $\mathcal{A}'$ follows the second distribution. If $\mathbf{B}$ is uniform, then $\mathbf{H}\mathbf{G} - \mathbf{B}$ is also uniform. As a result, the leftover hash lemma from Lemma 2.1 gives that $\mathbf{u}_i$ is within statistical distance $\frac{1}{2}\sqrt{q^d 2^{-H_\infty((\mathbf{v}'_i)_{\mathcal{P}_2})}}$ of the uniform, where $\mathbf{v}'_i$ is the subvector of $\mathbf{v}_i$ corresponding to the last $d+m_2$ entries. This is because we apply the leftover hash lemma on a uniform matrix, whereas our full matrix has a first block which is the identity. It thus yields that

$$\mathrm{Adv}[\mathcal{B}] \geq \delta - \frac{Q}{2}\sqrt{q^d 2^{-H_\infty((\mathbf{v}'_i)_{\mathcal{P}_2})}}.$$

In $\mathcal{P}_2$, $\mathbf{v}_{1,i}$ and $\mathbf{v}_{2,i}$ are sampled independently and therefore $H_\infty((\mathbf{v}'_i)_{\mathcal{P}_2}) = H_\infty(\mathscr{D}'_t) + H_\infty(\mathbf{G}^{-1}(U(\mathbb{Z}_q^d)))$, with $\mathscr{D}'_t$ being the marginal distribution of the last d entries of $(\mathbf{v}_1)_{\mathcal{P}_2}$. The definition of $\mathbf{G}^{-1}$ implies that the entropy of $H_\infty(\mathbf{G}^{-1}(U(\mathbb{Z}_q^d)))$ is exactly $d\log_2 q$. This is due to the fact that $\mathbf{G}^{-1}(\cdot)$ is a bijection between $\mathbb{Z}_q^d$ and $\mathbf{G}^{-1}(\mathbb{Z}_q^d)$, and thus preserves the entropy of its input. Under our LWE assumption, we then obtain

$$\delta \leq \varepsilon_{\mathrm{LWE}} + \frac{Q}{2} 2^{-H_\infty(\mathscr{D}'_t)/2}.$$

Combined with Equations (1) and (2), we get

$$\mathrm{Adv}_{\mathcal{P}_1,\mathcal{P}_2}[\mathcal{A}] \leq \varepsilon_{\mathrm{LWE}} + Q\left(2^{-\frac{1}{2}H_\infty(\mathscr{D}'_t)-1} + \frac{\varepsilon''}{M} + \frac{\varepsilon'(M+1)}{2M}\right),$$

as claimed. □

Theorem 3.1 proves that when (half of) $\mathscr{D}_t$ carries sufficient min-entropy, and that $\varepsilon, \varepsilon', \varepsilon''$ are negligible, then the output $\mathbf{v}$ of $\mathsf{SamplePre}$ is is indistinguishable from a distribution that is independent of the trapdoor $\mathbf{R}$, albeit conditioned on $[\mathbf{A}|\mathbf{HG} - \mathbf{AR}]\mathbf{v} = \mathbf{u} \bmod q\mathbb{Z}$. Since $\mathbf{AR} \bmod q\mathbb{Z}$ is generally made public, $\mathcal{P}_2$ acts as a simulator of $\mathcal{P}_1$ which does not require the trapdoor $\mathbf{R}$, a property we desire to have for trapdoor preimage sampling. Just like [LW15, Thm. 3.1], this improved simulation result carries over to an algebraic setting over number fields using [LW20, Cor. 5.9], at the expense of requiring low-splitting of the unramified prime q. The low-splitting is used to argue that $\mathbf{v} \bmod \mathfrak{q}$ carries enough entropy, where $\mathfrak{q}|qR$ and $\mathfrak{q} \neq R$. Typically, $\mathbf{v}_2 \bmod \mathfrak{q}$ carries at least $df \log_2 q$ bits of entropy, where $f = n/l$ and l the number of prime ideal factors of qR. Later, we use a modulus q that splits into 2 prime ideal factors in the power-of-two cyclotomic field of degree n.

3.3.4 Gaussian Instantiation. We can instantiate Theorem 3.1 with a Gaussian distribution on $\mathbf{v}_1$ for a fair comparison with previous results. This instantiation is also crucial in our aggregate signature scheme of Section 5 as we rely on the geometric properties of Gaussians to have a more compact aggregation. We thus choose $\mathscr{D}_r = U([-1, 1])$ for the trapdoor distribution, and we select $\mathscr{D}_s = \mathscr{D}_t = \mathcal{D}_{\mathbb{Z}^{m_1}, s}$ for the source and target distributions. For convenience, we write $\mathsf{SamplePre}(\mathbf{R}; \mathbf{A}', \mathbf{H}, \mathbf{u}_i, s)$ instead of specifying $\mathscr{D}_s$ and $\mathscr{D}_t$. In order to set s, we first derive the appropriate bound T on $\mathbf{Rv}_2$ with Lemma 2.4. Then, we choose a repetition rate $M > 1$ which defines the minimal slack $\alpha > 0$ so that $s = \alpha T$. This leads to the following corollary, which will be more convenient to use later.

Corollary 3.1. *Let λ, d, q, b, Q be positive integers with q prime. Let $m_1 = 2d$, $k = \lceil \log_b q \rceil$, $m_2 = dk$ and assume that $d \geq 5(\lambda + 4 + \log_2 Q)/\log_2 e$. We define $t_1 = \sqrt{(\lambda + 4 + \log_2 Q)/(\pi \log_2 e)}$ and $t_2 = \sqrt{(\lambda + 3 + \log_2 Q)/(\pi \log_2 e)}$, and $T = (b-1)\sqrt{m_2} \min(2\sqrt{m_1}, \sqrt{m_1} + \sqrt{m_2} + t_1)$. Let $\alpha > 0$, $M = \exp(\pi(\alpha^{-2} + 2t_2\alpha^{-1}))$, and finally $s = \alpha T$. Let $\mathbf{A}' \sim U(\mathbb{Z}_q^{d \times d})$, $\mathbf{R} \sim U([-1, 1]^{m_1 \times m_2})$ and $\mathbf{H} \in GL_d(\mathbb{Z}_q)$. We define $\mathcal{P}_1$ and $\mathcal{P}_2$ the same way as in Theorem 3.1 but where $\mathscr{D}_s, \mathscr{D}_t$ are replaced with $\mathcal{D}_{\mathbb{Z}^{m_1}, s}$. Then, it holds that the advantage of any PPT distinguisher $\mathcal{A}$ between $\mathcal{P}_1$ and $\mathcal{P}_2$ is at most*

$$Adv_{\mathcal{P}_1, \mathcal{P}_2}[\mathcal{A}] \leq \varepsilon_{\mathrm{LWE}} + Q\left(2^{-\frac{d \log_2 s + 1}{2}} + 2^{-(\lambda + 3 + \log_2 Q)} \frac{M+3}{2M}\right),$$

where $\varepsilon_{\mathrm{LWE}}$ is the hardness bound of $\mathrm{LWE}^{m_2}_{d,d,q,\mathscr{D}_r}$. In particular, if $\varepsilon_{\mathrm{LWE}} \leq 2^{-(\lambda+1)}$, then $Adv_{\mathcal{P}_1, \mathcal{P}_2}[\mathcal{A}] \leq 2^{-\lambda}$.

Proof. We simply have to verify that the conditions of Theorem 3.1 are met. First, because of the condition on d and the way we set t_1, we have $m_1 \geq 10(\lambda + 4 + \log_2 Q)/\log_2 e$ and Lemma 2.4 yields

$$\mathbb{P}_{\mathbf{R}, \mathbf{v}_2}[\|\mathbf{Rv}_2\|_2 > T] \leq 2^{-(\lambda + 4 + \log_2 Q)} + 2e^{-\pi t_1^2} = 2^{-(\lambda + 3 + \log_2 Q)} =: \varepsilon'.$$

Additionally, for $\mathbf{v}_1 \sim \mathcal{D}_{\mathbb{Z}^{m_1},s}$ and $\mathbf{y} = \mathbf{R}\mathbf{v}_2$ such that $\|\mathbf{y}\|_2 \leq T$, we have

$$\frac{\mathcal{D}_{\mathbb{Z}^{m_1},s}(\mathbf{v}_1)}{(\mathbf{y}+\mathcal{D}_{\mathbb{Z}^{m_1},s})(\mathbf{v}_1)} = \frac{\mathcal{D}_{\mathbb{Z}^{m_1},s}(\mathbf{v}_1)}{\mathcal{D}_{\mathbb{Z}^{m_1},s}(\mathbf{v}_1-\mathbf{y})} = \exp\left(\frac{\pi}{s^2}(\|\mathbf{y}\|_2^2 - 2\langle \mathbf{y},\mathbf{v}_1\rangle)\right).$$

By Lemma 2.3, it holds that $|\langle \mathbf{y},\mathbf{v}_1\rangle| \leq st_2\|\mathbf{y}\|_2$ except with probability at most $2e^{-\pi t_2^2} = 2^{-(\lambda+3+\log_2 Q)} = \varepsilon'$. Conditioned on $|\langle \mathbf{y},\mathbf{v}_1\rangle| \leq st_2\|\mathbf{y}\|_2$, we have

$$\begin{aligned}\frac{\mathcal{D}_{\mathbb{Z}^{m_1},s}(\mathbf{v}_1)}{(\mathbf{y}+\mathcal{D}_{\mathbb{Z}^{m_1},s})(\mathbf{v}_1)} &\leq \exp\left(\frac{\pi}{s^2}(\|\mathbf{y}\|_2^2 - 2t_2 s\|\mathbf{y}\|_2)\right)\\ &\leq \exp\left(\pi((T/s)^2 + 2t_2(T/s))\right)\\ &= \exp(\pi(\alpha^{-2} + 2t_2\alpha^{-1}))\\ &= M.\end{aligned}$$

We then obtain that

$$\mathbb{P}_{\mathbf{v}_1\sim\mathcal{D}_{\mathbb{Z}^{m_1},s}}[M(\mathbf{y}+\mathcal{D}_{\mathbb{Z}^{m_1},s})(\mathbf{v}_1) \geq \mathcal{D}_{\mathbb{Z}^{m_1},s}(\mathbf{v}_1)] \geq 1-\varepsilon',$$

and we can set $\varepsilon'' = \varepsilon' = 2^{-\lambda-3}/Q$. Note that the marginal distribution $\mathscr{D}'_t$ for $\mathscr{D}_t = \mathcal{D}_{\mathbb{Z}^{m_1},s}$ is exactly $\mathcal{D}_{\mathbb{Z}^d,s}$. Since $s \geq \eta_\delta(\mathbb{Z}^d)$ for some $\delta \in (0,1/2)$, Lemma 2.2 gives $H_\infty(\mathcal{D}_{\mathbb{Z}^d,s}) \geq d\log_2 s - 1$. It thus yields

$$2^{-\frac{1}{2}H_\infty(\mathcal{D}_{\mathbb{Z}^d,s})-1} \leq 2^{-\frac{d\log_2 s-1}{2}-1},$$

and in turn

$$\mathrm{Adv}_{\mathcal{P}_1,\mathcal{P}_2}[\mathcal{A}] \leq \varepsilon_{\mathrm{LWE}} + Q\left(2^{-\frac{d\log_2 s+1}{2}} + \varepsilon'\frac{M+3}{2M}\right),$$

by Theorem 3.1 as $\varepsilon' = \varepsilon''$.

Finally, because of our condition on d, which we use to set T, we have $2^{-\frac{d\log_2 s+1}{2}} \leq \varepsilon'$. It then holds that

$$\mathrm{Adv}_{\mathcal{P}_1,\mathcal{P}_2}[\mathcal{A}] \leq \varepsilon_{\mathrm{LWE}} + Q\varepsilon'(1+\frac{M+3}{2M}) \leq \varepsilon_{\mathrm{LWE}} + 3\cdot 2^{-(\lambda+3)}.$$

Assuming $\varepsilon_{\mathrm{LWE}} \leq 2^{-(\lambda+1)}$ gives $\mathrm{Adv}_{\mathcal{P}_1,\mathcal{P}_2}[\mathcal{A}] \leq 2^{-\lambda}$ as claimed. □

In this specific instantiation of [LW15] and Theorem 3.1 with Gaussian distributions, we only reach widths s which are larger than the ones from [MP12]. Indeed, in the latter, $\mathbf{v}_1$ was distributed according to a discrete Gaussian of width $s = \Theta(b\|\mathbf{R}\|_2) = \Theta(b(\sqrt{m_1}+\sqrt{m_2}))$, while here we obtain a width $s = \Theta(b\sqrt{m_2}(\sqrt{m_2}+\sqrt{m_1}))$. However, in the meantime, we drastically reduce the size of $\mathbf{v}_2$, which somewhat compensate for the increase in size of $\mathbf{v}_1$ for typical parameters. Although this may appear as irrelevant in standard applications of MP trapdoors at first glance, we show in Section 4 that it leads to interesting improvements in the size of preimages, and that it also finds advanced applications as that of Section 5 which were vacuous prior to our work.

4 Optimal Gadget Base and Sampler Performances

In the computational instantiation of MP trapdoors, the gadget base b is an important parameter to optimize over. Since the base defines the length of the gadget matrix $m_2 = d\lceil \log_b q \rceil$, choosing a larger base results in lower dimensional vectors, at the expense of a larger norm. As the norm only impacts the bitsize logarithmically while the dimension impacts it linearly, one could think that the optimal choice for b is around $\sqrt{q}$, thus resulting in $m_2 = 2d$, smaller preimages and in turn smaller signatures. We however discuss here that the optimal base actually depends on the preimage sampler. We illustrate our discussion with the instructive example of GPV signatures [GPV08] with MP trapdoors. We compare the original sampler of [MP12] (thereafter called MP signatures) with the elliptic sampler we introduced Section 3.2 (later called ellMP signatures), and that of [LW15] (recalled in Algorithm 3.1) with our new simulation from Corollary 3.1 (later called LW signatures). We consider this simple example as the security analysis and parameter estimate are quite straightforward. Other more complex constructions would require a brand new security analysis when changing the sampler, especially with the LW sampler which has a very different structure. Nevertheless, from this simple example alone, we witness interesting improvement factors on the size of preimages from our analyses. This represents a step towards concrete practicality of constructions based on MP trapdoors.

GPV Signature. We briefly describe the signature from [GPV08] with MP trapdoors in their computational instantiation based on M-LWE. The secret key $\mathbf{R}$ is drawn from $U(S_1^{m_1 \times m_2})$ with $m_1 = 2d$, and the public key is composed of $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}'] \in R_q^{d \times m_1}$ and $\mathbf{B} = \mathbf{AR} \bmod qR$. As described before, the signature of a message $\mathbf{m} \in \{0,1\}^*$ consists of a short preimage $\mathbf{v} = [\mathbf{v}_1^T | \mathbf{v}_2^T]^T \in R^{m_1+m_2}$ satisfying

$$[\mathbf{A} | \mathbf{G} - \mathbf{B}]\mathbf{v} = \mathcal{H}(\mathbf{m}) \bmod qR.$$

Since the matrix $\mathbf{A}$ has $\mathbf{I}_d$ as its first block, we can use similar tricks as for example [PFH+20,EFG+22,ETWY22] to reduce the signature size. The GPV signature now consists of $(\mathbf{v}_{1,2}, \mathbf{v}_2)$, where $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$, because $\mathbf{v}_{1,1}$ is determined by the verification equation as $\mathbf{v}_{1,1} = \mathcal{H}(\mathbf{m}) - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G} - \mathbf{B})\mathbf{v}_2$.

Choosing the Gadget Base. For a given base, the minimal Gaussian parameter needed for MP signatures $\mathbf{v}$ is $s = r\sqrt{b^2+3}\sqrt{\|\mathbf{R}\|_2^2 + 1}$, where r is a randomized rounding factor around $4-5$, and $\|\mathbf{R}\|_2$ can be bounded heuristically by $\sqrt{nm_1} + \sqrt{nm_2} + t$ for a slack $t \approx 7$. The bitsize of a signature is thus

$$|\mathsf{sig}_{\mathsf{MP}}| = |\mathbf{v}_{1,2}| + |\mathbf{v}_2| = nd(1 + \lceil \log_b q \rceil)\lceil \log_2 s \log_2 \lambda \rceil. \tag{3}$$

For ellMP signatures, we introduce an asymmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$ and thus have two Gaussian parameters $s_1 = r\sqrt{2}\sqrt{b^2+3}\|\mathbf{R}\|_2$ and $s_2 = s_1/\|\mathbf{R}\|_2$. The bitsize of a signature is

$$|\mathsf{sig}_{\mathsf{ellMP}}| = |\mathbf{v}_{1,2}| + |\mathbf{v}_2| = nd\lceil \log_2 s_1 \log_2 \lambda \rceil + nd\lceil \log_b q \rceil \lceil \log_2 s_2 \log_2 \lambda \rceil. \tag{4}$$

Finally, for the sampler from Algorithm 3.1, the Gaussian parameter for $\mathbf{v}_1$ is given by $s = \alpha\|\mathbf{R}\|_2(b-1)\sqrt{nm_2}$ where the slack α defines the repetition rate M. The bitsize of a signature is

$$|\mathsf{sig}_{\mathsf{LW}}| = |\mathbf{v}_{1,2}| + |\mathbf{v}_2| = nd\lceil \log_2 s \log_2 \lambda \rceil + nd\lceil \log_b q \rceil \lceil \log_2 b \rceil. \tag{5}$$

We already see that as opposed to Equations (3) and (4), the size of $\mathbf{v}_2$ for LW signatures in Equation (5) is roughly $nd \log_2 q$ independently of the choice of b. However, s increases with b which means that a larger base b would result in larger signatures. The opposite phenomenon happens for MP signatures as we observe that the function mapping b to $|\mathsf{sig}_{\mathsf{MP}}|$ is roughly non-increasing in typical parameter settings. Hence, based on this sole metric, larger bases b would give smaller signatures, and the optimal choice would therefore be $b = \lceil \sqrt{q} \rceil$. For ellMP signatures however, the asymmetry of the preimages places it in between MP and LW signatures. Typically, $|\mathsf{sig}_{\mathsf{ellMP}}|$ is non-increasing up to an inflexion point that is slightly smaller than that of MP signatures but that is still larger than $\lceil \sqrt{q} \rceil$. So the asymmetry introduced by our optimization does not impact the choice of the gadget base by much.

However, as illustrated in the tables below, the choice of the optimal base is not as straightforward as the formula above might lead us to think. One must indeed take into account the impact of the parameters on the underlying computational assumptions. Indeed, in the security proof, one needs to argue that simulated signatures lead to programmed random oracle responses which are close to uniform. To do so, we use the simulation result from Corollary 3.1 (or its equivalent for the old sampling procedure for the MP and ellMP samplers) in the module setting. As such, we need to consider parameters that ensure the M-LWE$_{n,d,d,q,U(S_1)}$ problem is hard. For a fair estimate, we aim at $\lambda + \log_2 m_2$ bits of security for M-LWE, as the pseudorandomness of $[\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$ is argued under the M-LWE assumption with m_2 secrets. The security proof is then concluded by a reduction to M-SIS$_{n,d,m_1+m_2,q,\beta}$ where $\beta \geq \|\mathbf{v} - \mathbf{v}^*\|_2$ for two preimages $\mathbf{v}, \mathbf{v}^*$. It yields $\beta = 2s\sqrt{nm_1 + nm_2}$ for MP signatures, $\beta = \sqrt{4s_1^2 nm_1 + 4s_2^2 nm_2}$ for ellMP signatures, and $\beta = \sqrt{4s^2 nm_1 + (b-1)^2 nm_2}$ for LW signatures. For MP signatures, the bound β is dominated by the bottom part $\mathbf{v}_2$ as $m_2 \geq m_1$. It thus makes sense to increase the base in order to reduce the dimension of m_2 and thus have balanced contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ to the M-SIS bound β. On the contrary, for ellMP and LW signatures, the asymmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$ due to the sampler reduces the size of $\mathbf{v}_2$ and thence re-balances the contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ in the bound β. Typically, for LW signatures, β is already dominated by $\mathbf{v}_1$ for $b = 2$, and increasing b will only enlarge the gap between the contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ to the M-SIS bound and inherently decrease the security.

Estimates. We now give performance estimates for MP, ellMP and LW signatures with parameters achieving $\lambda = 128$ bits of security for the GPV signature, using the Core-SVP methodology with sieving SVP oracle. For that, we fix the randomized rounding factor $r = 5.4$, and the spectral norm slack $t = 7$, and

rejection sampling slack $\alpha = 8$ (leading to a repetition rate of $M \approx 73$). We use $Q = 2^{40}$ as the maximal number of emitted signatures per key. We then find the appropriate dimension d and modulus q to achieve the security target while minimizing the signature size. Although our goal is also to optimize over the gadget base, we give the performance for several choices of base to show the overall trends described above.

The values of $\lambda_{\text{M-LWE}}$ and $\lambda_{\text{M-SIS}}$ correspond to the reached security of $\text{M-LWE}_{n,d,d,q,U(S_1)}$ and $\text{M-SIS}_{n,d,m_1+m_2,q,\beta}$ respectively. The estimates are given in Tables 4.1, 4.2 and 4.3. When the base is said to be $q^{1/k}$, we actually consider $b = \lceil q^{1/k} \rceil$ to have an integer base for which the gadget dimension is $m_2 = dk$. The optimal sizes and parameters are highlighted in the tables. Using the base $b = \sqrt{q}$ impacts the M-SIS bound too drastically, and parameters need to be increased to compensate the security accordingly. In particular, one has to ensure that the infinity norm of the M-SIS solution is smaller than q to avoid trivial solutions.

	$\lambda_{\text{M-LWE}}$	$\lambda_{\text{M-SIS}}$	q	d	s	$\lvert\mathbf{v}_{1,2}\rvert$	$\lvert\mathbf{v}_2\rvert$	$\lvert\mathsf{sig}_{\mathsf{MP}}\rvert$
$b = 2$	239	144	$\approx 2^{15.2}$	5	2655	2.34	37.50	39.84
$b = 4$	233	150	$\approx 2^{15.5}$	5	3461	2.34	18.75	21.09
$b = q^{1/5}$	216	147	$\approx 2^{16.7}$	5	7661	2.50	12.50	15.00
$b = q^{1/3}$	181	131	$\approx 2^{19.7}$	5	56804	**2.97**	**8.91**	**11.88**
$b = q^{1/2}$	194	154	$\approx 2^{26.7}$	7	6616938	5.69	11.37	17.06

Table 4.1. Parameter and size estimates of MP signatures using different bases b. The sizes are expressed in KB. The ring degree is $n = 256$.

	$\lambda_{\text{M-LWE}}$	$\lambda_{\text{M-SIS}}$	q	d	s_1	s_2	$\lvert\mathbf{v}_{1,2}\rvert$	$\lvert\mathbf{v}_2\rvert$	$\lvert\mathsf{sig}_{\mathsf{ellMP}}\rvert$
$b = 2$	178	134	$\approx 2^{15.5}$	4	3372	19	1.88	16.00	17.88
$b = 4$	173	130	$\approx 2^{16}$	4	4570	31	1.88	9.00	10.88
$b = q^{1/5}$	207	161	$\approx 2^{17.4}$	5	11797	86	2.66	7.81	10.47
$b = q^{1/3}$	174	136	$\approx 2^{20.4}$	5	94702	792	**3.13**	**6.09**	**9.22**
$b = q^{1/2}$	188	154	$\approx 2^{27.4}$	7	11926700	94109	5.91	8.75	14.66

Table 4.2. Parameter and size estimates of ellMP signatures using different bases b. The sizes are expressed in KB. The ring degree is $n = 256$.

These estimates indeed reflect the expected behavior, namely MP signatures becomes much more compact with larger bases while LW signatures are most efficient for smaller bases. They also show that in the context of GPV signatures, the ellMP sampler leads to more compact preimages than the MP sampler. Finally, our new analysis of Algorithm 3.1 leads to signatures that are more compact than with both the MP and ellMP samplers. Other signature de-

	$\lambda_{\text{M-LWE}}$	$\lambda_{\text{M-SIS}}$	q	d	s	$\lvert\mathbf{v}_{1,2}\rvert$	$\lvert\mathbf{v}_2\rvert$	$\lvert\mathsf{sig}_{\mathsf{LW}}\rvert$
$b = 2$	195	157	$\approx 2^{22.5}$	6	376491	**4.13**	**4.31**	**8.44**
$b = 4$	188	151	$\approx 2^{23.2}$	6	645772	4.31	4.50	8.81
$b = q^{1/5}$	167	134	$\approx 2^{25.6}$	6	3576993	4.69	5.62	10.31
$b = q^{1/3}$	167	137	$\approx 2^{30.3}$	7	90206170	6.56	7.22	13.78
$b = q^{1/2}$	162	138	$\approx 2^{40.3}$	9	90202905475	11.25	11.81	23.06

Table 4.3. Parameter and size estimates of LW signatures using different bases b. The sizes are expressed in KB. The ring degree is $n = 256$.

signs [DM14,BFRS18,dPLS18,BEP+21,LNPS21,LNP22,dPK22,JRS22] may benefit from using the sampler from [LW15]. However, the choice of the optimal sampler depends on the application and the security analysis. For GPV signatures, $\mathbf{v}_1$ and $\mathbf{v}_2$ contribute somewhat similarly to the M-SIS bound. In more complex designs, the M-SIS bound may already be dominated by $\mathbf{v}_1$, in which case the LW sampler may not be relevant as it increases the size of $\mathbf{v}_1$ while reducing that of $\mathbf{v}_2$.

5 A Lattice-Based Aggregate Signature Scheme

As concrete application of how we can leverage the asymmetry of the preimage resulting from our new analyses, we construct the first lattice-based aggregate signature that supports public aggregation and that is more efficient than the naive concatenation of individual signatures. It in particular shows that the LW sampler from Algorithm 3.1 improved as described in Section 3.3 can unlock new signature designs. We start by recalling the definition of aggregate signature schemes in Section 5.1, before presenting our construction in Section 5.2. Then, we prove the security of our scheme in the aggregate chosen-key model coined by Boneh et al. [BGLS03] in Section 5.3. Finally, we dedicate Section 5.4 to discussing the performance of our scheme.

5.1 Aggregate Signature Schemes

An aggregate signature is a regular signature scheme $\{\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}\}$ which also enables public aggregation of different signatures on different messages and under different signing keys. The regular signature is thus completed with two algorithms $\mathsf{AggSign}$ and $\mathsf{AggVerify}$. The former takes as input a sequence of messages $(\mathbf{m}_i)_{i\in[N]}$, of public keys $(\mathsf{pk}_i)_{i\in[N]}$ and of signatures $(\mathsf{sig}_i)_{i\in[N]}$ of said messages under the corresponding keys, and outputs a single signature $\mathsf{sig}_{\mathsf{agg}}$. The $\mathsf{AggVerify}$ algorithm then takes the same inputs except that it gets $\mathsf{sig}_{\mathsf{agg}}$ instead of the individual signatures, and returns 1 if the aggregate signature is valid and 0 otherwise. An aggregate signature scheme is expected to be correct, i.e., honestly generated signatures and aggregate signatures verify using Verify and

$\mathsf{AggVerify}$ respectively, and secure in a security model introduced by [BGLS03] which we recall in Section 5.3.

The goal of aggregate signatures is to perform batch verification of several independent signatures, albeit sharing the same public parameters. The naive solution is to define $\mathsf{sig}_{\mathsf{agg}}$ as the concatenation of the $(\mathsf{sig}_i)_{i\in[N]}$ and perform verification individually but the resulting construction is meaningless, except perhaps to show that aggregate signatures trivially exist. In practice, we are therefore interested in aggregate signature schemes that perform better than the naive concatenation.

As explained in Section 1.1, several aggregate signatures gathering such features have been proposed in the classical setting, but it was yet open to propose a post-quantum construction. A first attempt over lattices was proposed by Döroz et al. [DHSS20], but had major drawbacks either in performance (MMSA) or security ($\mathsf{MMSAT}/\mathsf{MMSATK}$), and was based on a non-standard assumption called Vandermonde-SIS (or Partial Fourier Recovery). Boudgoust and Roux-Langlois [BR21] then proposed another lattice-based aggregate signature based on the FSwA paradigm, which unfortunately ended up being larger than the trivial concatenation. One explanation of this lack of compression is the half aggregation and the peculiarities of aggregate signatures which in the end make the parameters slightly worse than for the standalone signature. In particular, FSwA signatures are composed of two parts $(\mathsf{sig}_1, \mathsf{sig}_2)$ and only one of them can be aggregated, i.e., the aggregate signature is of the form $\mathsf{sig}_{\mathsf{agg}} = (\mathsf{sig}_1, (\mathsf{sig}_{2,i})_{i\in[N]})$ where $(\mathsf{sig}_{1,i}, \mathsf{sig}_{2,i})_{i\in[N]}$ are the signatures to be aggregated. Unfortunately, one needs larger parameters to prove the security of the aggregate signature scheme. As a result the size of the non-aggregated part $\mathsf{sig}_{2,i}$ becomes larger than the size of a full FSwA signature with the smaller parameters. Hence, $\mathsf{sig}_{\mathsf{agg}}$ is always larger than the concatenation of standalone signatures in the case of [BR21], regardless of the value of N.

We now present a lattice-based aggregate signature scheme that supports public aggregation, whose security is proven in the aggregate chosen-key model based on standard (module) lattice assumptions, and that performs better than the naive solution. This answers positively to the open problem left by Boudgoust et al. in [BR21], and provides, to the best of our knowledge, the first post-quantum aggregate signature combining all such features.

5.2 Our Construction

Our aggregate signature scheme is based on the GPV hash-and-sign framework [GPV08], with MP trapdoors [MP12] and the preimage sampling algorithm of [LW15] presented in Section 3 with our new parameter analysis. We present our scheme over module lattices.

As explained in Section 4, the combination of the GPV signature and MP trapdoors produces signatures $\mathsf{sig} = \mathbf{v}$ on messages $\mathbf{m}$ by sampling the preimage $\mathbf{v}$ of $\mathcal{H}(\mathbf{m})$ by $[\mathbf{A}|\mathbf{G} - \mathbf{AR}] \bmod q$. The function $\mathcal{H}$ is modeled by a random oracle, the matrix $\mathbf{A}$ is uniformly random and part of the public key, while $\mathbf{R}$ is a short matrix constituting the secret key. The matrix $\mathbf{B} = \mathbf{AR}$ is also

part of the public key. For different users, each user i would have a set of keys $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{B}_i = \mathbf{A}_i\mathbf{R}_i)$ and $\mathsf{sk}_i = \mathbf{R}_i$. An intuitive way of aggregating signatures sig_i is to sum them, but this becomes tricky when the public matrices involved in verification, i.e., $[\mathbf{A}_i|\mathbf{G}-\mathbf{B}_i]$, are all different. We can however force all the $\mathbf{A}_i$ to be the same matrix $\mathbf{A}$ for all i, making sure $\mathbf{A}$ is honestly generated, i.e., without embedding an illicit trapdoor. This can for example be done by setting $\mathbf{A}$ as the hash of some public parameters. Each user would thus share the same $\mathbf{A}$ and would have their own public key $\mathbf{B}_i = \mathbf{A}\mathbf{R}_i$. Hence, by summing the verification equations, we would obtain $\mathbf{A}\cdot\sum_{i\in[N]}\mathbf{v}_{1,i}+\sum_{i\in[N]}(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i} = \sum_{i\in[N]}\mathcal{H}(\mathbf{m}_i)$. The aggregate signature could then be $(\sum_i \mathbf{v}_{1,i}, (\mathbf{v}_{2,i})_i)$, meaning we would only be aggregating the $\mathbf{v}_{1,i}$ and providing the individual $\mathbf{v}_{2,i}$.

As in the previous attempts [DHSS20,BR21], it seems difficult to achieve full aggregation due to the fact that $\mathbf{v}_{2,i}$ faces $\mathbf{B}_i$, which must differ for every user. As a result, the bit size of the first half $\sum_i \mathbf{v}_{1,i}$ would grow logarithmically with N, while that of the second half $(\mathbf{v}_{2,i})_i$ would grow linearly with N. Similarly to FSwA signatures, as described in Section 5.1, if the increased complexity of aggregate signature security results in $\mathbf{v}_{2,i}$ being larger that a full MP signature $(\mathbf{v}_1, \mathbf{v}_2)$, the aggregate signature scheme would be vacuous. Fortunately, based on our new assessment, the preimage sampler recalled in Section 3 moves the bulk of the signatures in the $\mathbf{v}_{1,i}$ while minimizing the size of $\mathbf{v}_{2,i}$ which makes the concatenation of the $\mathbf{v}_{2,i}$ minimal. It therefore amortizes the linear cost of the aggregate signature, and each $\mathbf{v}_{2,i}$ in the aggregate signature stays sufficiently below the size of a full LW signature to allow for relevant compression.

Unfortunately, this aggregate signature is not secure as it is. Indeed, one can note that the user j can produce a forged aggregate signature on behalf of the set of users $1, \ldots, N$ as follows:

1. Select a set of messages $\mathbf{m}_i$, for $i \in [N]$.
2. Select $\mathbf{v}_{2,i}$, for $i \neq j$, distributed as in a normal signature.
3. Compute $\mathbf{v}_{2,j}$ such that $\mathbf{G}\mathbf{v}_{2,j} = -\sum_{i\neq j}(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i} + \sum_{i\in[N]}\mathcal{H}(\mathbf{m}_i)$.
4. Set $\mathbf{v}_1 = \mathbf{R}_j\mathbf{v}_{2,j}$.

The resulting aggregate signature $(\mathbf{v}_1, (\mathbf{v}_{2,i})_i)$ is indeed valid on $(\mathbf{m}_i)_i$ under public keys $(\mathbf{B}_i)_i$ since

$$\begin{aligned}\mathbf{A}\cdot\mathbf{v}_1 + \sum_{i\in[N]}(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i} &= \mathbf{A}\cdot\mathbf{v}_1 + (\mathbf{G}-\mathbf{B}_j)\mathbf{v}_{2,j} + \sum_{i\neq j}(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i}\\ &= \mathbf{G}\mathbf{v}_{2,j} + \sum_{i\neq j}(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i}\\ &= \sum_{i\in[N]}\mathcal{H}(\mathbf{m}_i).\end{aligned}$$

Intuitively, the problem stems from the fact that the rogue signer is able to compute its own signature after seeing/selecting the other components. It can thus use its own trapdoor to select a preimage that will cancel all these components. To solve this problem, we rely on a countermeasure reminiscent of the one

used against rogue key attacks. We tweak the verification equation with small random weights e_i that deterministically depend on the full set $\{(\mathbf{m}_i, \mathbf{v}_{2,i}, \mathbf{B}_i)\}_i$. This therefore forces the adversary to commit to each $\mathbf{v}_{2,i}$ before seeing the verification equation it must satisfy, which thwarts the previous attack.

However, if we follow the standard approach where $e_i \leftarrow \mathcal{H}(\mathbf{B}_1, \mathbf{v}_{2,1}, \mathbf{m}_1, \ldots, \mathbf{B}_N, \mathbf{v}_{2,N}, \mathbf{m}_N, i)$ for some hash function $\mathcal{H}$, we will end up with the same problem as in [BR21]: we could only ensure unforgeability for the last signature (the one generated under public key $\mathbf{B}_N$). This has led the authors in [BR21] to use a specific security model, where the challenge key must necessarily be the last one, but the real-world security assurances provided by this model are questionable. Informally, the problem is related to the forking lemma: at some point in the security proof we need to rewind and change the weight e_j associated with the challenge public key $\mathbf{B}_j$. However, the proof works only if e_j is the last weight to be queried to the random oracle, hence the restriction in the model of [BR21]. Otherwise, the adversary could change the other weights after the rewinding, which would completely invalidate the proof strategy. Here, we stress that one cannot simply run the simulation several times until this event (e_j is the last queried weight) happens because j is known to the adversary (it is the index corresponding to the challenge public key). Therefore, an adversary could systematically initiate its queries with e_j, leading this probabilistic approach to fail.

We show that we can circumvent this issue at almost no cost by generating the small elements e_i in two steps. Concretely, we first compute f as the output of hash function $\mathcal{H}_f$ taking as input $\{\mathbf{B}_j, \mathbf{v}_{2,j}, \mathbf{m}_j\}_j$. The output space is denoted by F but there are no restrictions on it because f is then fed to another random oracle. The only constraint is that $|F|$ must be exponential in the security parameter to avoid simple guessing or collision-finding attacks. Then, each e_i is generated as the output of another hash function $\mathcal{H}_e$ run on (f, i). Here, the output of the random oracle shall be small polynomials. We typically use ternary polynomials e_i with fixed Hamming weight, i.e., in $\mathcal{C} = \{e \in S_1 : \|e\|_1 = w\}$. Intuitively, this resorting to two successive random oracles $\mathcal{H}_f, \mathcal{H}_e$ enables the simulation to anticipate the weight queries and, more importantly, to control their order. This way, we can rely on the forking lemma without placing any contrived restrictions on the model, at the cost of only one hash evaluation for the whole aggregate signature.

The sampler from [LW15] given in Algorithm 3.1 can be instantiated so that it samples the $\mathbf{v}_{1,i}$ close to a Gaussian distribution, which is the object of Corollary 3.1. Although [LW15] can be used for a broader class of distributions such as uniform over a hypercube, the properties of Gaussian distributions lead to tighter verification bounds and in turn a smaller M-SIS bounds and thus smaller parameters. More precisely, the weighted sum $\mathbf{v}_1 = \sum_{i\in[N]} e_i \mathbf{v}_{1,i}$ follows a Gaussian distribution, and the tail bound thus gives $\|\mathbf{v}_1\|_2 \leq w \cdot \sqrt{N} \cdot s\sqrt{nm_1}$. For other distribution, one would use the triangle inequality and get $\|\mathbf{v}_1\|_2 \leq \sum_{i\in[N]} \|e_i\|_1 \|\mathbf{v}_{1,i}\|_2 \leq w \cdot N \cdot B$ where B would be the norm bound on each $\mathbf{v}_{1,i}$

for a single signature. The dependency in N is therefore optimized in the case of Gaussian distributions.

Finally, as in Section 4, we can consider the matrix $\mathbf{A}$ in Hermite Normal Form, i.e., $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ with $\mathbf{A}' \sim U(R_q^{d\times d})$. If each $\mathbf{v}_{1,i}$ is parsed as $[\mathbf{v}_{1,1,i}^T|\mathbf{v}_{1,2,i}^T]^T$ with $\mathbf{v}_{1,1,i}, \mathbf{v}_{1,2,i} \in R^d$, this allows us to only aggregate the $\mathbf{v}_{1,2,i}$ as $\mathbf{v}_{1,2} = \sum_{i\in[N]} e_i\mathbf{v}_{1,2,i}$. The other part, i.e., $\mathbf{v}_{1,1} = \sum_{i\in[N]} e_i\mathbf{v}_{1,1,i}$ can be recovered during verification as

$$\mathbf{v}_{1,1} = \sum_{i\in[N]} e_i\mathcal{H}(\mathbf{m}_i) - \mathbf{A}'\mathbf{v}_{1,2} - \sum_{i\in[N]} e_i(\mathbf{G}-\mathbf{B}_i)\mathbf{v}_{2,i}.$$

Although this does not have a tremendous impact on the aggregate signature size when N is large, as the bulk of it is due to the concatenation of the $\mathbf{v}_{2,i}$, it leads to a more compact signature and gives a fair comparison with concatenated LW signatures.

The Scheme. In what follows, we work over the $2n$-th cyclotomic ring denoted by R for n a power of two, as defined in Section 2.3. Although we have seen that the optimal base for the sampler from [LW15] seems to be $b = 2$, we present the scheme for an arbitrary b and optimize over it to give the most efficient parameter sets. The aggregate signature is described by Algorithms 5.1 to 5.6.

Algorithm 5.1: Setup

Input: Security parameter λ, Maximal number of signers N.

1. Choose a positive integers d, q, w, b with q prime and $q = 5 \bmod 8$.
2. $\mathcal{C} \leftarrow \{e \in S_1 : \|e\|_1 = w\}$. ▷ Hash space for weights, such that $|\mathcal{C}| \geq 2^{2\lambda}$
3. $k \leftarrow \lceil \log_b q \rceil$.
4. $(m_1, m_2) \leftarrow (2d, dk)$.
5. $\mathbf{G} = \mathbf{I}_d \otimes [1 \cdots b^{k-1}] \in R_q^{d\times dk}$. ▷ Gadget vector
6. $t \leftarrow \sqrt{(3\lambda/2 + 4 + \log_2 Q)/(\pi \log_2 e)}$. ▷ $t \approx 7$
7. Choose $\alpha > 0$. ▷ Rejection Sampling Slack
8. $M \leftarrow \exp(\pi(\alpha^{-2} + 2t\alpha^{-1}))$. ▷ Repetition rate
9. $s \leftarrow \max(\alpha(b-1)\sqrt{nm_2}(\sqrt{nm_1} + \sqrt{nm_2} + t), w\sqrt{2}\eta_\varepsilon(\mathbb{Z}^{nm_1}))$. ▷ Width
10. $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$.

Output: $\mathsf{pp} = (\mathbf{A}'; \mathbf{G}; \lambda, N, n, q, d, m_1, m_2, w, k, s, M)$.

Algorithm 5.2: KeyGen

Input: Public parameters pp as in Algorithm 5.1.

1. $\mathbf{R} \hookleftarrow U(S_1^{m_1\times m_2})$
2. $\mathbf{B} \leftarrow [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR \in R_q^{d\times m_2}$

Output: $\mathsf{pk} = \mathbf{B}$, and $\mathsf{sk} = \mathbf{R}$. ▷ pp stored with pk for simplicity

Algorithm 5.3: Sign

Input: Secret key sk, Message $\mathbf{m} \in \{0,1\}^*$, Public key pk.

1. **if** $(\mathbf{m}, \mathbf{v})$ is stored **then** look-up $\mathbf{v}$
2. **else** $\mathbf{v} \leftarrow \mathsf{SamplePre}(\mathbf{R}; \mathbf{A}', \mathbf{I}_d, \mathcal{H}(\mathbf{m}), s)$. ▷ Algorithm 3.1
3. Store $\mathbf{v}$. Parse $\mathbf{v}$ as $[\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T|\mathbf{v}_2^T]^T$ with $\mathbf{v}_{1,1}, \mathbf{v}_{1,2} \in R^d$ and $\mathbf{v}_2 \in R^{m_2}$.

Output: $\mathsf{sig} = (\mathbf{v}_{1,2}, \mathbf{v}_2)$.

Algorithm 5.4: Verify

Input: Public key pk, Message $\mathbf{m} \in \{0,1\}^*$, Signature sig.

1. $\mathbf{v}_{1,1} \leftarrow \mathcal{H}(\mathbf{m}) - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G} - \mathbf{B})\mathbf{v}_2 \in R^d$
2. $\mathbf{v}_1 \leftarrow [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T \in R^{m_1}$.
3. $b \leftarrow (\|\mathbf{v}_1\|_2 \leq s\sqrt{nm_1}) \wedge (\mathbf{v}_2 \in T_b^{m_2})$

Output: b. ▷ $b = 1$ if valid, 0 otherwise

Algorithm 5.5: AggSign

Input: Public keys $(\mathbf{B}_i)_{i\in[N]}$, Signatures $(\mathbf{v}_{1,2,i}, \mathbf{v}_{2,i})_{i\in[N]}$, Messages $(\mathbf{m}_i)_{i\in[N]}$

1. $f \leftarrow \mathcal{H}_f(\mathbf{B}_1, \mathbf{v}_{2,1}, \mathbf{m}_1, \ldots, \mathbf{B}_N, \mathbf{v}_{2,N}, \mathbf{m}_N) \in F$ ▷ $|F| \geq |\mathcal{C}| \geq 2^{2\lambda}$
2. $\forall i \in [N], e_i \leftarrow \mathcal{H}_e(f, i) \in \mathcal{C}$.
3. $\mathbf{v}_{1,2} \leftarrow \sum_{i\in[N]} e_i \mathbf{v}_{1,2,i}$.

Output: $\mathsf{sig}_{\mathsf{agg}} = (\mathbf{v}_{1,2}, (\mathbf{v}_{2,i})_{i\in[N]})$.

Algorithm 5.6: AggVerify

Input: Public keys $(\mathbf{B}_i)_{i\in[N]}$, Aggregate Signature $(\mathbf{v}_{1,2}, (\mathbf{v}_{2,i})_{i\in[N]})$, Messages $(\mathbf{m}_i)_{i\in[N]}$

1. $f \leftarrow \mathcal{H}_f(\mathbf{B}_1, \mathbf{v}_{2,1}, \mathbf{m}_1, \ldots, \mathbf{B}_N, \mathbf{v}_{2,N}, \mathbf{m}_N) \in F$
2. $\forall i \in [N], e_i \leftarrow \mathcal{H}_e(f, i) \in \mathcal{C}$.
3. $\mathbf{v}_{1,1} \leftarrow \sum_{i\in[N]} e_i \mathcal{H}(\mathbf{m}_i) - \mathbf{A}'\mathbf{v}_{1,2} - \sum_{i\in[N]} e_i(\mathbf{G} - \mathbf{B}_i)\mathbf{v}_{2,i}$
4. $\mathbf{v}_1 \leftarrow [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T \in R^{m_1}$.
5. $b_1 \leftarrow (\|\mathbf{v}_1\|_2 \leq ws\sqrt{N \cdot nm_1})$.
6. $b_2 \leftarrow (\forall i \in [N], \mathbf{v}_{2,i} \in T_b^{m_2})$

Output: $b_1 \wedge b_2$. ▷ 1 if valid, 0 otherwise

We give prove the correctness of our scheme in the following theorem.

Theorem 5.1 (Correctness). *The aggregate signature scheme (Setup, KeyGen, Sign, Verify, AggSign, AggVerify) described in Section 5.2 is correct. Formally, for all security parameters λ and number of signers N, the following hold.*

Single signature correctness. *For all* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$, *for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, *for all* $\mathbf{m} \in \{0,1\}^*$,

$$\mathbb{P}[\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, \mathsf{Sign}(\mathsf{sk}, \mathbf{m}; \mathsf{pk})) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

Aggregate signature correctness. *For all* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$, *for all* $i \in [N]$ *and for all* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\mathbf{m}_i \in \{0,1\}^*$, $\mathsf{sig}_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, \mathbf{m}_i; \mathsf{pk}_i)$,

$$\mathbb{P}[\mathsf{AggVerify}(\mathbf{PK}, \mathsf{AggSign}(\mathbf{PK}, \mathbf{SIG}, \mathbf{M}), \mathbf{M}) = 1] \geq 1 - \mathsf{negl}(\lambda),$$

where $\mathbf{PK} = (\mathsf{pk}_i)_{i\in[N]}$, $\mathbf{SIG} = (\mathsf{sig}_i)_{i\in[N]}$ *and* $\mathbf{M} = (\mathbf{m}_i)_{i\in[N]}$.

Proof. We first look at the single signature correctness. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$, $(\mathbf{B}, \mathbf{R}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\mathbf{m} \in \{0,1\}^*$, and $(\mathbf{v}_{1,2}, \mathbf{v}_2) \leftarrow \mathsf{Sign}(\mathbf{R}, \mathbf{m}; \mathbf{B})$. We reconstruct $\mathbf{v}_{1,1} \leftarrow \mathcal{H}(\mathbf{m}) - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G} - \mathbf{B})\mathbf{v}_2$ and $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$. It thus holds that $[\mathbf{v}_1^T | \mathbf{v}_2^T]^T$ was obtained using $\mathsf{SamplePre}$. Using the parameters of Algorithm 5.1,similarly as in the proof Theorem 3.1, Lemma 2.5 gives that $\mathbf{v}_1$ is within statistical distance at most $1 \cdot (\varepsilon'/M + \varepsilon''(M+1)/(2M)) \leq 2^{-3\lambda/2-2}/Q$ of $\mathcal{D}_{R^{m_1},s}$, where $\varepsilon', \varepsilon''$ are as in Corollary 3.1 satisfying $\varepsilon', \varepsilon'' \leq 2^{-3\lambda/2-3}/Q$.

Notice that in the correctness we look at one signature which explains the factor 1 and not Q. Lemma 2.3 then yields

$$\mathbb{P}[\mathsf{Verify}(\mathbf{B}, \mathbf{m}, \mathbf{v}) = 1] \geq 1 - 2^{-3\lambda/2-2}/Q - 2^{-2nm_1} = 1 - \mathsf{negl}(\lambda).$$

Additionally, by construction it holds that $\mathbf{v}_2 \in T_b^{m_2}$.

Let us now investigate the correctness of our aggregate signature. Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, N)$, and for all $i \in [N]$ let $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\mathbf{m}_i \in \{0,1\}^*$, $\mathsf{sig}_i \leftarrow \mathsf{Sign}(\mathsf{sk}_i, \mathbf{m}_i; \mathsf{pk}_i)$. Let $\mathsf{sig}_{\mathsf{agg}} \leftarrow \mathsf{AggSign}(\mathbf{PK}, \mathbf{SIG}, \mathbf{M})$ and parse it as $(\mathbf{v}_{1,2}, (\mathbf{v}_{2,i})_{i\in[N]})$. From the single signature correctness above, we directly have that $b_2 = 1$, namely that $\mathbf{v}_{2,i} \in T_b^{m_2}$ for all $i \in [N]$.

We reconstruct $\mathbf{v}_{1,1} \leftarrow \sum_{i\in[N]} e_i \mathcal{H}(\mathbf{m}_i) - \mathbf{A}'\mathbf{v}_{1,2} - \sum_{i\in[N]} e_i(\mathbf{G}-\mathbf{B})\mathbf{v}_{2,i}$ and $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$. Since the signatures were honestly generated, it holds that $\mathbf{v}_1 = \sum_{i\in[N]} e_i \mathbf{v}_{1,i}$ where $[\mathbf{v}_{1,i}^T | \mathbf{v}_{2,i}^T]^T$ was obtained using $\mathsf{SamplePre}$.

We now look at the norm bound on $\mathbf{v}_1$. The idea is that $\mathbf{v}_1$ behaves as a discrete Gaussian over a lattice that depends on the weights e_i and its covariance depends on the size of the e_i. Using the Gaussian tail bound of Lemma 2.3 yields the correct bound. We now give more details. First, since $\mathbf{v}_1$ is a weighted sum of discrete Gaussian vectors, Lemma 2.6 yields

$$\Delta(\sum_{i\in[N]} e_i \mathcal{D}_{R^{m_1},s}, \mathcal{D}_{\sum_{i\in[N]} e_i R^{m_1}, \sqrt{\mathbf{S}}}) \leq \mathsf{negl}(\lambda),$$

where $\mathbf{S} = \mathbf{I}_{m_1} \otimes \sum_{i\in[N]} s^2 M_\tau(e_i) M_\tau(e_i)^T$, as long as the Gaussian width verifies $s \geq \sqrt{2}\eta_\varepsilon(\mathbb{Z}^{nm_1}) \cdot \max_{i\in[N]} \|M_\tau(e_i)\|_2$. Due to the specific form of $M_\tau(e_i)$ as described in Section 2.3, it holds by e.g. [BJRW23, Lem. 2.2] that $\|M_\tau(e_i)\|_2 \leq \|\tau(e_i)\|_1 = w$. The condition thus becomes $s \geq w\sqrt{2}\eta_\varepsilon(\mathbb{Z}^{nm_1})$, which is encompassed by our parameter choice. Then, using the fact that each e_i has weight $w \neq 0$, it holds that $e_i \neq 0$ in the field K and in turn that all the $M_\tau(e_i)$ are invertible. As a result, the final covariance matrix $\mathbf{S}$ is positive definite. Using [GMPW20, Lem. 2.3], we obtain that

$$\mathcal{D}_{\sum_{i\in[N]} e_i R^{m_1}, \sqrt{\mathbf{S}}} = \sqrt{\mathbf{S}} \mathcal{D}_{\sqrt{\mathbf{S}}^{-1} \sum_{i\in[N]} e_i R^{m_1}, 1},$$

and we can therefore apply Lemma 2.3 and get

$$\begin{aligned}
&\mathbb{P}_{\mathbf{v}_1 \sim \mathcal{D}_{\sum_i e_i R^{m_1}, \sqrt{\mathbf{S}}}} \left[\|\mathbf{v}_1\|_2 > \left\|\sqrt{\mathbf{S}}\right\|_2 \sqrt{nm_1} \right] \\
&\qquad = \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\sqrt{\mathbf{S}}^{-1} \sum_i e_i R^{m_1}, 1}} \left[\left\|\sqrt{\mathbf{S}}\mathbf{x}\right\|_2 > \left\|\sqrt{\mathbf{S}}\right\|_2 \sqrt{nm_1} \right] \\
&\qquad \leq \mathbb{P}_{\mathbf{x} \sim \mathcal{D}_{\sqrt{\mathbf{S}}^{-1} \sum_i e_i R^{m_1}, 1}} \left[\|\mathbf{x}\|_2 > \sqrt{nm_1} \right] \\
&\qquad \leq 2^{-2nm_1},
\end{aligned}$$

where the first inequality follows by inclusion of events. We now only need to bound $\left\|\sqrt{\mathbf{S}}\right\|_2$. The latter corresponds to $\sqrt{\lambda_{\max}(\mathbf{S})}$ which itself equals $\sqrt{\lambda_{\max}(\mathbf{S}')}$

with $\mathbf{S}' = s^2 \sum_{i\in[N]} M_\tau(e_i) M_\tau(e_i)^T$, and where $\lambda_{\max}$ denotes the largest eigenvalue. Recalling from Section 2.3 that $M_\tau = \mathbf{P}^H M_\sigma \mathbf{P}$ with $\mathbf{P}$ a unitary matrix, a standard calculation yields

$$\mathbf{S}' = s^2 \mathbf{P}^H \mathrm{diag}\left(\sum_{i\in[N]} |\sigma_1(e_i)|^2, \ldots, \sum_{i\in[N]} |\sigma_n(e_i)|^2 \right) \mathbf{P},$$

where the σ_i are the individual field embeddings. It thus proves that

$$\lambda_{\max}(\mathbf{S}') = s^2 \max_{k\in[n]} \sum_{i\in[N]} |\sigma_k(e_i)|^2.$$

For all (k, i), we have $|\sigma_k(e_i)| \leq \|\sigma(e_i)\|_\infty = \|M_\sigma(e_i)\|_2$. By [BJRW23, Lem. 2.3], it gives $\|M_\sigma(e_i)\|_2 = \|M_\tau(e_i)\| \leq \|\tau(e_i)\|_1 = w$. As a result, we obtain $\lambda_{\max}(\mathbf{S}') \leq s^2 N w^2$. Combining the rejection sampling, the weighted sum of Gaussians, the tail bound and the spectral bound on $\lambda_{\max}(\mathbf{S})$, it proves that

$$\mathbb{P}_{\mathbf{v}_1}[\|\mathbf{v}_1\|_2 > ws\sqrt{N \cdot nm_1}] \leq N \cdot 2^{-3\lambda/2-2}/Q + 2^{-2nm_1} + \mathsf{negl}(\lambda),$$

thus proving that $b_1 = 1$ except with negligible probability, as $N \ll Q$ and $N = \mathsf{poly}(\lambda)$. It then yields

$$\begin{aligned} \mathbb{P}[\mathsf{AggVerify}(\mathbf{PK}, \mathbf{M}, \mathsf{sig}_{\mathsf{agg}}) = 1] &\geq 1 - N \cdot 2^{-3\lambda/2-2}/Q - 2^{-2nm_1} - \mathsf{negl}(\lambda) \\ &= 1 - \mathsf{negl}(\lambda), \end{aligned}$$

concluding the proof. □

5.3 Security

The *aggregate chosen-key* security model introduced by Boneh et al. [BGLS03] captures the idea that an adversary cannot produce an aggregate signature on behalf of N users, even if it colludes with (at most) $N-1$ of them. The adversary is given a challenge public key pk and the ability to query signatures on this key, and is asked to produce $N-1$ keys pk_i as well as an aggregate signature $\mathsf{sig}_{\mathsf{agg}}$ that verifies with these N public keys. We formally define this model by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$ in three stages.

Setup Stage. $\mathcal{B}$ runs Setup and KeyGen to obtain pp, pk, and sk. It then gives pp and pk to $\mathcal{A}$.

Query Stage. $\mathcal{A}$ queries signatures on at most Q messages $\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(Q)}$, which are answered by $\mathcal{B}$ returning $\mathsf{sig}^{(i)} \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{m}^{(i)}; \mathsf{pk})$.

Forgery Stage. $\mathcal{A}$ eventually provides a forgery $((\mathsf{pk}_i)_{i\in[N]}, (\mathbf{m}_i)_{i\in[N]}, \mathsf{sig}_{\mathsf{agg}})$.

The adversary wins the game if (1) there exists an $i^* \in [N]$ such that $\mathsf{pk}_{i^*} = \mathsf{pk}$, (2) for all $i \in [Q]$, $\mathbf{m}_{i^*} \neq \mathbf{m}^{(i)}$, and (3) $\mathsf{AggVerify}((\mathsf{pk}_i)_{i\in[N]}, \mathsf{sig}_{\mathsf{agg}}, (\mathbf{m}_i)_{i\in[N]}) = 1$. The adversary's advantage is defined as $\mathrm{Adv}[\mathcal{A}] = \mathbb{P}[\mathcal{A} \text{ wins}]$, where the probability is over all the random coins. We say that the aggregate signature scheme

is secure in the aggregate chosen-key model if for all probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $\mathrm{Adv}[\mathcal{A}]$ is negligible in the security parameter λ.

We note that in [BGLS03], the challenge key is set to be pk_1. In the context of their construction in bilinear groups, this can be assumed without loss of generality because the order of the signatures that are aggregated does not matter. In our case, each (half) signature $\mathbf{v}_{1,i}$ is multiplied by a weight $e_i = \mathcal{H}_e(f, i)$ which depends on the position i and also the order of the signatures because of $f = \mathcal{H}_f(\mathbf{B}_1, \mathbf{v}_{2,1}, \mathbf{m}_1, \ldots, \mathbf{B}_N, \mathbf{v}_{2,N}, \mathbf{m}_N)$. These weights are necessary in the lattice setting to avoid the attack we described in Section 5.2. As a result, in the security proof, the challenger has to guess the position i^* of the challenge key in order to exploit the forgery to break the underlying computational assumption.

Theorem 5.2 (Security). *The aggregate signature scheme (Setup, KeyGen, Sign, Verify, AggSign, AggVerify) described in Section 5.2 is secure in the aggregate chosen-key model under the* M-SIS *and* M-LWE *assumptions. More formally, for any PPT adversary $\mathcal{A}$ against the aggregate chosen-key security, it holds that*

$$Adv[\mathcal{A}] \leq N \cdot \left(2\varepsilon_{\text{M-LWE}} + \frac{Q_e}{|\mathcal{C}|} + \sqrt{Q_e \varepsilon_{\text{M-SIS}}}\right) + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda),$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bounds of $\text{M-LWE}^{m_2}_{n,d,d,q,U(S_1)}$, and $\varepsilon_{\text{M-SIS}}$ is that of $\text{M-SIS}_{n,d,m_1+m_2,q,\beta}$ with $\beta = \sqrt{(2w(\sqrt{N}+1)s\sqrt{nm_1})^2 + (4w(b-1)\sqrt{nm_2})^2}$.

Proof. We proceed by a sequence of games that we prove indistinguishable from the aggregate chosen-key game. In the final game, we use the general forking lemma in order to deduce a solution of M-SIS. We first denote by Q the maximal number of signature queries, and by Q_e the maximal number of queries to $\mathcal{H}_e$.

Game G_0. We change the original aggregate chosen-key game by programming the random oracles in a certain way. The challenger $\mathcal{B}$ starts by sampling $i^+ \hookleftarrow U([N])$, which later acts as a guess on the position of the challenge key in the forgery. $\mathcal{B}$ is also provided with some random inputs $h_j \hookleftarrow U(\mathcal{C})$ for all $j \in [Q_e]$. Additionally, $\mathcal{B}$ keeps four tables $\mathcal{T}_s, \mathcal{T}_f, \mathcal{T}_e, \mathcal{T}_m$ that will be used to store the corresponding queries, and which are all empty at the outset of the game. Finally, it further stores an index j_e, initially set to 0.

Setup. $\mathcal{B}$ computes $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\mathbf{B}, \mathbf{R}) = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$. It then sends pp, pk to $\mathcal{A}$.

Queries to $\mathcal{H}$. On input $\mathbf{m} \in \{0,1\}^*$ given by $\mathcal{A}$, $\mathcal{B}$ first checks whether $\mathbf{m}$ is already stored in $\mathcal{T}_m$. If so, it directly outputs the $\mathbf{u}$ from $\mathcal{T}_m$ corresponding to $\mathbf{m}$. If not, it samples $\mathbf{u} \hookleftarrow U(R_q^d)$, stores $(\mathbf{m}, \mathbf{u})$ in $\mathcal{T}_m$ and sends $\mathbf{u}$ to $\mathcal{A}$.

Queries to $\mathcal{H}_f$. On input $(\mathbf{B}_i, \mathbf{v}_{2,i}, \mathbf{m}_i)_{i\in[N]}$ given by $\mathcal{A}$, $\mathcal{B}$ first checks whether it already appears in $\mathcal{T}_f$. If so, it directly outputs the f in $\mathcal{T}_f$ corresponding to the input. If not, it samples $f \hookleftarrow U(F)$, stores $((\mathbf{B}_i, \mathbf{v}_{2,i}, \mathbf{m}_i)_{i\in[N]}, f)$ in $\mathcal{T}_f$ and sends f to $\mathcal{A}$. Additionally, for all $i \in [N] \setminus \{i^+\}$, $\mathcal{B}$ samples $e_i \hookleftarrow U(\mathcal{C})$ and stores (f, i, e_i) in $\mathcal{T}_e$.

Queries to $\mathcal{H}_e$. On input (f, i) given by $\mathcal{A}$, $\mathcal{B}$ first checks whether it already appears in $\mathcal{T}_e$. If so, it outputs the e_i from $\mathcal{T}_e$ corresponding to (f, i). If (f, i) does not appear in $\mathcal{T}_e$, then either f does not appear in $\mathcal{T}_f$ or $i = i^+$. Without loss of generality, we can assume that f has previously been obtained by a query to $\mathcal{H}_f$, and therefore we necessarily have $i = i^+$. Then, $\mathcal{B}$ increments j_e to $j_e + 1$ and sends h_{j_e} to $\mathcal{A}$. It also stores (f, i^+, h_{j_e}) in $\mathcal{T}_e$. Notice that $\mathcal{H}_e(f, i^+)$ is therefore set after all the other $\mathcal{H}_e(f, i)$ for $i \neq i^+$.

Signature queries. On input $\mathbf{m}$, $\mathcal{B}$ first checks if $\mathbf{m}$ appears in $\mathcal{T}_s$. If so, it outputs the $\mathbf{v}$ from $\mathcal{T}_s$ corresponding to $\mathbf{m}$. If not, it proceeds as follows. $\mathcal{B}$ checks if $\mathbf{m}$ is in $\mathcal{T}_m$. If not, it samples $\mathbf{u} \hookleftarrow U(R_q^d)$ and stores $(\mathbf{m}, \mathbf{u})$ in $\mathcal{T}_m$. Otherwise, it gets the corresponding syndrome $\mathbf{u}$. Then, it runs the legitimate signing algorithm Sign with $\mathsf{sk}, \mathsf{pk}, \mathsf{pp}$ by just replacing $\mathcal{H}(\mathbf{m})$ by $\mathbf{u}$, namely sampling $\mathbf{v} = (\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \mathbf{v}_2) \leftarrow \mathsf{SamplePre}(\mathbf{R}; \mathbf{A}', \mathbf{I}_d, \mathbf{u}, s)$. It then stores $(\mathbf{m}, \mathbf{v})$ in $\mathcal{T}_s$ and sends $(\mathbf{v}_{1,2}, \mathbf{v}_2)$ to $\mathcal{A}$.

Forgery. Eventually, $\mathcal{A}$ outputs $((\mathsf{pk}_i)_{i\in[N]}, (\mathbf{m}_i)_{i\in[N]}, \mathsf{sig}_{\mathsf{agg}})$ to $\mathcal{B}$ such that there exists $i^* \in [N]$ satisfying $\mathsf{pk}_{i^*} = \mathsf{pk}$, that $\mathbf{m}_{i^*}$ was not part of the signing queries, and such that $\mathsf{AggVerify}((\mathsf{pk}_i)_{i\in[N]}, \mathsf{sig}_{\mathsf{agg}}, (\mathbf{m}_i)_{i\in[N]}) = 1$. If these conditions are not met, then $\mathcal{B}$ outputs $(0, \perp)$. From now on, we assume that these conditions are met, which happens with probability $\mathrm{Adv}[\mathcal{A}]$ as everything is correctly distributed. Then, if $i^* \neq i^+$, then $\mathcal{B}$ also outputs $(0, \perp)$. Since i^+ is completely independent of the view of $\mathcal{A}$ as all the random oracle queries are identical as in the standard game, this happens with probability $1/N$. If $f = \mathcal{H}_f((\mathsf{pk}_i, \mathbf{v}_{2,i}, \mathbf{m}_i)_{i\in[N]})$ was not queried, then $\mathcal{A}$ would have had to guess the correct value of f to obtain the weights e_i, and thus the signature would verify with probability at most $1/|F|$. Noting that $1/|F| = \mathsf{negl}(\lambda)$, it would entail a negligible advantage for $\mathcal{A}$. So we assume that f has been queried. Similarly, if $\mathcal{H}_e(f, i^+)$ was not queried, then the probability that $\mathsf{AggVerify}$ passes is at most $1/|\mathcal{C}|$ as $\mathcal{A}$ would have had to guess the value of e_{i^+}. Since $1/|\mathcal{C}| = \mathsf{negl}(\lambda)$, then such an adversary $\mathcal{A}$ would have a negligible advantage. So we further assume, without loss of generality that $\mathcal{H}_e(f, i^+)$ was queried and is equal to some h_j for some counter index j. Then, $\mathcal{B}$ outputs (j, out) with $\mathsf{out} = ((\mathsf{pk}_i)_{i\in[N]}, (\mathbf{m}_i)_{i\in[N]}, \mathsf{sig}_{\mathsf{agg}}, (\mathcal{H}_e(f, i))_{i\in[N]}))$. Further, we let p_k denote the probability that $\mathcal{B}$ does not output $(0, \perp)$ in game G_k. Here, we have

$$p_0 = \frac{1}{N}\mathrm{Adv}[\mathcal{A}]. \tag{6}$$

Game G_1. This game is identical to game G_0 except in the way signatures are generated. Instead, $\mathcal{B}$ simulates signatures without resorting to sk by using the simulator from Corollary 3.1. We thus change the way queries to $\mathcal{H}$ and signing queries are handled.

Queries to $\mathcal{H}$. On input $\mathbf{m} \in \{0,1\}^*$ given by $\mathcal{A}$, $\mathcal{B}$ first checks whether $\mathbf{m}$ is already stored in $\mathcal{T}_m$. If so, it directly outputs the $\mathbf{u}$ from $\mathcal{T}_m$ corresponding to $\mathbf{m}$. If not, it samples $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{m_1}, s}$, $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$, sets $\mathbf{v} = [\mathbf{v}_1^T | \mathbf{v}_2^T]^T \in R^{m_1+m_2}$ and computes $\mathbf{u} = [\mathbf{I}_d | \mathbf{A}' | \mathbf{G} - \mathbf{B}]\mathbf{v} \bmod qR$. It rejects such a $\mathbf{v}, \mathbf{u}$ with

probability $1-1/M$ and repeats the procedure until $\mathbf{v}, \mathbf{u}$ is kept. Then, $\mathcal{B}$ stores $(\mathbf{m}, \mathbf{u})$ in $\mathcal{T}_m$ and $(\mathbf{m}, \mathbf{v})$ in $\mathcal{T}_s$. It then sends $\mathbf{u}$ to $\mathcal{A}$.

Signature queries. On input $\mathbf{m} \in \{0,1\}^*$ given by $\mathcal{A}$, $\mathcal{B}$ first checks whether $\mathbf{m}$ is already stored in $\mathcal{T}_s$. If so, it directly outputs the $\mathbf{v}$ from $\mathcal{T}_s$ corresponding to $\mathbf{m}$. If not, it means that $\mathcal{H}$ was never queried on $\mathbf{m}$. In this case, $\mathcal{B}$ performs the query to $\mathcal{H}(\mathbf{m})$ on its own as above and fills $\mathcal{T}_m$ with $(\mathbf{m}, \mathbf{u})$ and $\mathcal{T}_s$ with $(\mathbf{m}, \mathbf{v})$. It then sends $\mathbf{v}$ to $\mathcal{A}$.

The simulation result of [LW15, Thm. 3.1] which we overhauled in Theorem 3.1 applies to the Gaussian case as stated in Corollary 3.1. Hence, the latter, extended to the module setting as explained in Section 3.3.3, yields that

$$|p_0 - p_1| \leq \varepsilon_{\text{M-LWE}} + Q \cdot 2^{-3\lambda/2-1-\log_2 Q} = \varepsilon_{\text{M-LWE}} + \mathsf{negl}(\lambda). \tag{7}$$

Game G_2. Since sk is no longer used in game G_1, we define G_2 to be identical to G_1 except in the setup stage.

Setup. $\mathcal{B}$ computes $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and samples $\mathbf{B}' \hookleftarrow U(R_q^{d \times m_2})$. It then computes $\mathbf{B} \leftarrow \mathbf{G} - \mathbf{B}'$ and sets $\mathsf{pk} \leftarrow \mathbf{B}$. It then sends pp, pk to $\mathcal{A}$.

Since $\mathbf{B}'$ is uniform, then so is $\mathbf{B}$. By the $\text{M-LWE}^{m_2}_{n,d,d,q,U(S_1)}$ assumption, $[\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$ in game G_1 is $\varepsilon_{\text{M-LWE}}$-indistinguishable from $\mathbf{B}$ in game G_2. As a result, it holds that

$$|p_1 - p_2| \leq \varepsilon_{\text{M-LWE}}. \tag{8}$$

Forking. We now aim at bounding p_2, using the general forking lemma recalled in Lemma 2.7. We use the forking algorithm $\mathcal{F}_\mathcal{B}$ of Algorithm 2.1 around $\mathcal{B}$ and we will invoke Lemma 2.7. The input generator IG is defined by outputting $\overline{\mathbf{A}} = [\mathbf{I}_d|\mathbf{A}'|\mathbf{B}']$ where $[\mathbf{A}'|\mathbf{B}'] \hookleftarrow U(R_q^{d \times (m_1 - d + m_2)})$ and pp honestly generated (where $\mathbf{A}'$ is the same matrix as the one in pp). For clarity, we denote by $\mathbf{A}$ the matrix $[\mathbf{I}_d|\mathbf{A}']$. We call acc the accepting probability of $\mathcal{B}$, i.e., $\mathsf{acc} = p_2$, and frk the forking probability from Lemma 2.7. Hence, with probability frk, the two calls to $\mathcal{B}$, and in turn $\mathcal{A}$ (which are both oblivious to the fact they are being rewound), return (j, out) and (j', out') with $j = j' \neq 0$ and $h_j \neq h'_j$. The output of $\mathcal{F}_\mathcal{B}$ is in this case $(1, \mathsf{out}, \mathsf{out}')$. We now use $\mathsf{out}, \mathsf{out}'$ to construct a solution to M-SIS on the matrix $\overline{\mathbf{A}}$.

By definition of the forking, we have that the random coins are the same up to the forking index j. As a result, $(f, i^+) = (f', i^+)$ and $e_{i^+} = h_j \neq h'_j = e'_{i^+}$. Because $f = f'$, this implies that $\mathsf{pk}_i = \mathsf{pk}'_i$, $\mathbf{v}_{2,i} = \mathbf{v}'_{2,i}$ and $\mathbf{m}_i = \mathbf{m}'_i$ for all $i \in [N]$. Additionally, due to the fact that e_{i^+} is set before all the e_i in the queries to $\mathcal{H}_e$, we have that $e_i = e'_i$ for all $i \neq i^+$. Then, since $\mathsf{sig}_{\mathsf{agg}}$ and $\mathsf{sig}'_{\mathsf{agg}}$ both verify, by definition of the reconstructed vectors $\mathbf{v}_{1,1}, \mathbf{v}'_{1,1}$ in Algorithm 5.6 and $\mathbf{v}_1, \mathbf{v}'_1$, we have

$$\begin{aligned}
\mathbf{A}\mathbf{v}_1 + \sum_{i \in [N]} e_i(\mathbf{G} - \mathbf{B}_i)\mathbf{v}_{2,i} &= \sum_{i \in [N]} e_i \mathcal{H}(\mathbf{m}_i) \bmod qR \\
\mathbf{A}\mathbf{v}'_1 + \sum_{i \in [N]} e'_i(\mathbf{G} - \mathbf{B}'_i)\mathbf{v}'_{2,i} &= \sum_{i \in [N]} e'_i \mathcal{H}(\mathbf{m}'_i) \bmod qR,
\end{aligned}$$

such that $\|\mathbf{v}_1\|_2, \|\mathbf{v}'_1\|_2 \leq ws\sqrt{Nnm_1}$. We call $\Delta e = e_{i^+} - e'_{i^+}$. With the prior observations, combining the above equations gives

$$\mathbf{A}(\mathbf{v}_1 - \mathbf{v}'_1) + \Delta e \cdot (\mathbf{G} - \mathbf{B})\mathbf{v}_{2,i^+} = \Delta e \cdot \mathcal{H}(\mathbf{m}_{i^+}) \bmod qR$$

We note that $\mathbf{m}_{i^+}$ was not queried for a signature, but it must have been queried to $\mathcal{H}$ (otherwise $\mathcal{A}$ would have had a negligible advantage to begin with). Hence, $\mathcal{T}_s$ contains an entry $(\mathbf{m}_{i^+}, \mathbf{v}'')$ where $\mathbf{v}''$ was generated as in game G_2. Then, $\mathbf{v}''$ verifies $\mathbf{A}\mathbf{v}''_1 + (\mathbf{G} - \mathbf{B})\mathbf{v}''_2 = \mathcal{H}(\mathbf{m}_{i^+}) \bmod qR$. We then obtain

$$\mathbf{A}(\mathbf{v}_1 - \mathbf{v}'_1 - \Delta e \cdot \mathbf{v}''_1) + \Delta e \cdot (\mathbf{G} - \mathbf{B})(\mathbf{v}_{2,i^+} - \mathbf{v}''_2) = \mathbf{0} \bmod qR,$$

which can be written $\overline{\mathbf{A}}\mathbf{x} = \mathbf{0} \bmod qR$ for

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_1 - \mathbf{v}'_1 \\ \Delta e \cdot \mathbf{v}_{2,i^+} \end{bmatrix} - \Delta e \cdot \mathbf{v}'' \in R^{m_1+m_2}.$$

The adversary $\mathcal{A}$ does not know $\mathbf{v}''$ but only $\overline{\mathbf{A}}\mathbf{v}'' \bmod qR$ which takes $2^{nd\log_2 q}$ possible values. By [DORS08, Lem. 2.2], the entropy of $\mathbf{v}''$ given $\overline{\mathbf{A}}\mathbf{v}'' \bmod qR$ is at least $H_\infty(\mathbf{v}'') - nd\log_2 q$. Since $\mathbf{v}''$ is sampled by the simulator, it holds that $\mathbf{v}''_1 \sim \mathcal{D}_{R^{m_1},s}$ and $\mathbf{v}''_2 \sim \mathbf{G}^{-1}(U(R_q^d))$. As a result, $H_\infty(\mathbf{v}'') = H_\infty(\mathcal{D}_{R^{m_1},s}) + nd\log_2 q$. Then, by Lemma 2.2, we have that $H_\infty(\mathcal{D}_{R^{m_1},s}) \geq nm_1\log_2 s - 1$ as $s \geq \eta_\delta(R^{m_1})$ for some negligible $\delta > 0$. We thus obtain that the entropy of $\mathbf{v}''$ given $\overline{\mathbf{A}}\mathbf{v}'' \bmod qR$ is at least $nm_1\log_2 s - 1 \gg 4\lambda$, and then that $\mathbf{x} = \mathbf{0}$ only with negligible probability. Finally, we have

$$\begin{aligned}
\|\mathbf{x}\|_2 &\leq \sqrt{(\|\mathbf{v}_1\|_2 + \|\mathbf{v}'_1\|_2 + \|\Delta e\|_1\|\mathbf{v}''_1\|_2)^2 + (\|\Delta e\|_1 \cdot (\left\|\mathbf{v}_{2,i^+}\right\|_2 + \|\mathbf{v}''_2\|_2))^2} \\
&\leq \sqrt{(2w \cdot (\sqrt{N}+1) \cdot s\sqrt{nm_1})^2 + (2w \cdot 2(b-1)\sqrt{nm_2})^2} \\
&= \beta,
\end{aligned}$$

except with probability $2^{-2nm_1} \ll 2^{-4\lambda}$ that is due to Lemma 2.3. Therefore, $\mathbf{x}$ is a solution to $\text{M-SIS}_{n,d,m_1+m_2,q,\beta}$ except with negligible probability. Since we assumed that the hardness bound of the latter was $\varepsilon_{\text{M-SIS}}$, it thus hold that

$$\mathsf{frk} \leq \varepsilon_{\text{M-SIS}} + \mathsf{negl}(4\lambda) \tag{9}$$

Combining Equation (9) with the result from the general forking lemma, we get

$$p_2 = \mathsf{acc} \leq \frac{Q_e}{|\mathcal{C}|} + \sqrt{Q_e(\varepsilon_{\text{M-SIS}} + \mathsf{negl}(4\lambda))}.$$

We can assume without loss of generality that $Q_e \leq 2^\lambda$, and recalling that $\mathcal{C}$ is chosen so that $|\mathcal{C}| \geq 2^{2\lambda}$, it holds $Q_e/|\mathcal{C}| = \mathsf{negl}(\lambda)$. Combined with Equations (6), (7), and (8), we get

$$\text{Adv}[\mathcal{A}] \leq N \cdot \left(2\varepsilon_{\text{M-LWE}} + \frac{Q_e}{|\mathcal{C}|} + \sqrt{Q_e\varepsilon_{\text{M-SIS}}}\right) + \mathsf{negl}(\lambda),$$

as claimed. □

5.4 Performance Evaluation

We now evaluate the performance of our aggregate signature compared to the naive concatenation. For that we define the compression rate as

$$\mathsf{compression\ rate} = 100 \cdot \left(1 - \frac{|\mathsf{sig_{agg}}|}{|\mathsf{concatenation}|}\right) \%.$$

However, to obtain a fair comparison, we cannot simply compare the concatenation of signatures produced by Algorithm 5.3 with the aggregate signature output by Algorithm 5.5. Indeed, in the case of a mere concatenation, the parameters used in Algorithm 5.3 would not be optimal, one would instead use those for single GPV signatures, as described in Section 4. We thus compare below the size of an aggregate signature with the concatenation of signatures generated with better parameters, tailored to the single signature use-case. Concretely, although we use the same ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, where $n = 256$, we select the optimal parameters from Table 4.3, that is $q \approx 2^{22.5}$, $d = 6$, $b = 2$, $s \approx 376491$ for single signatures, leading to signature size of $|\mathsf{sig}| = 69120$ bits ≈ 8.44 KB. Hence, the concatenation of N signatures results in a naive aggregate signature of $|\mathsf{concatenation}| = N \cdot 69120$ bits.

We estimate the aggregate signature size for different values of N ranging from $N = 10$ to $N = 1200$. The bit-size of the aggregate signature is given by

$$|\mathsf{sig_{agg}}| = n(m_1 - d)\left\lceil \log_2(ws\sqrt{N}\log_2 \lambda)\right\rceil + N \cdot nd\lceil \log_b q\rceil\lceil \log_2 b\rceil.$$

The parameters of our scheme are set according to Setup (Algorithm 5.1) with $Q = 2^{40}$, where q, d and b are selected to guarantee sufficient security for the underlying M-SIS$_{n,d,m_1+m_2,q,\beta}$ and M-LWE$^{m_2}_{n,d,d,q,U(S_1)}$ problems while minimizing the aggregate signature size. Since the parameters increase with N (typically the bound β), the values of q and d will naturally depend on N accordingly. Hence, when N increases the modulus q and rank d need to be increased to preserve the security of the scheme, which results in lower compression rates. The higher N gets, the more we would have to increase q and d, and we thus expect that passed a certain threshold N the concatenation would become better than our aggregate signature. Nevertheless, in practical use cases of aggregate signatures the number of signers stays in the low hundreds which in our case offer a $10-15\%$ compression rate compared to the naive concatenation, as shown in Table 5.1.

Acknowledgments. This work has received a French government support managed by the National Research Agency in the ASTRID program, under the national project AMIRAL with reference ANR-21-ASTR-0016, and in the MobiS5 project with reference ANR-18-CE-39-0019-02 MobiS5. We warmly thank Vadim Lyubashevsky for helpful discussions on the Lyubashevsky-Wichs sampler. We also thank David Pointcheval for his insight on the use of the forking lemma, Katharina Boudgoust for her constructive feedback on the use of Gaussians in aggregate signatures, and Nicholas Genise for interesting discussions.

N	5	10	50	100	500	1000	1200
$\lvert\mathsf{concat}\rvert$	42.2	84.4	421.9	843.8	4218.8	8437.5	10125
(d, q)	$(6, 2^{22.5})$	$(6, 2^{22.5})$	$(6, 2^{22.5})$	$(6, 2^{22.5})$	$(6, 2^{22.5})$	$(6, 2^{22.5})$	$(6, 2^{22.5})$
$\lvert\mathsf{sig}_{\mathsf{agg}}\rvert$	45.5	80.5	363.6	713.6	3732.8	7670.7	9464.4
(d, q)	$(7, 2^{29.9})$	$(7, 2^{30.3})$	$(7, 2^{31.3})$	$(7, 2^{31.7})$	$(7, 2^{33.9})$	$(7, 2^{34.8})$	$(7, 2^{35.2})$
Comp. Rate	0%	4.59%	13.82%	15.43%	11.52%	9.09%	6.52%

Table 5.1. Comparison estimates of our aggregate signature and the concatenation of LW signatures over module lattices. Sizes of $\lvert\mathsf{concat}\rvert$ and $\lvert\mathsf{sig}_{\mathsf{agg}}\rvert$ are expressed in KB.

References

ABB+20. E. Alkim, P. S. L. M. Barreto, N. Bindel, J. Krämer, P. Longa, and J. E. Ricardini. The lattice-based digital signature scheme qtesla. In *ACNS*, 2020.

AKSY22. S. Agrawal, E. Kirshanova, D. Stehlé, and A. Yadav. Practical, round-optimal lattice-based blind signatures. In *CCS*, 2022.

Ban93. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Math. Ann.*, 1993.

BDK+18. J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *EuroS&P*, 2018.

BEP+21. P. Bert, G. Eberhart, L. Prabel, A. Roux-Langlois, and M. Sabt. Implementation of lattice trapdoors on modules and applications. In *PQCrypto*, 2021.

BFRS18. Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-sis/lwe based signature and IBE. In *PQCrypto*, 2018.

BGLS03. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, 2003.

BGP22. K. Boudgoust, E. Gachon, and A. Pellet-Mary. Some easy instances of ideal-svp and implications on the partial vandermonde knapsack problem. In *CRYPTO*, 2022.

BJRW20. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. Towards classical hardness of module-lwe: The linear rank case. In *ASIACRYPT*, 2020.

BJRW23. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module learning with errors with short distributions. *J. Cryptol.*, 36:1, 2023.

BN06. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *CCS*, 2006.

BNN07. M. Bellare, C. Namprempre, and G. Neven. Unrestricted aggregate signatures. In *ICALP*, 2007.

BR21. K. Boudgoust and A. Roux-Langlois. Non-interactive half aggregate signatures based on module lattices - a first attempt. *IACR Cryptol. ePrint Arch.*, page 263, 2021.

BTT22. C. Boschini, A. Takahashi, and M. Tibouchi. Musig-l: Lattice-based multi-signature with single-round online phase. In *CRYPTO*, 2022.

DHSS20. Y. Doröz, J. Hoffstein, J. H. Silverman, and B. Sunar. MMSAT: A scheme for multimessage multiuser signature aggregation. *IACR Cryptol. ePrint Arch.*, page 520, 2020.

DKL+18. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018.

DLP14. L. Ducas, V. Lyubashevsky, and T. Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT*, 2014.

DM14. L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *CRYPTO*, 2014.

DORS08. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 2008.

dPK22. R. del Pino and S. Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In *CRYPTO*, 2022.

dPLS18. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *CCS*, 2018.

EFG+22. T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, and Y. Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In *EUROCRYPT*, 2022.

ETWY22. Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter hash-and-sign lattice-based signatures. In *CRYPTO*, 2022.

GM18. Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *EUROCRYPT*, 2018.

GMPW20. N. Genise, D. Micciancio, C. Peikert, and M. Walter. Improved discrete gaussian and subgaussian analysis for lattice cryptography. In *Public Key Cryptography*, 2020.

GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

HKW15. S. Hohenberger, V. Koppula, and B. Waters. Universal signature aggregators. In *EUROCRYPT*, 2015.

HW18. S. Hohenberger and B. Waters. Synchronized aggregate signatures from the RSA assumption. In *EUROCRYPT*, 2018.

ISO13. ISO/IEC. 20008-2:2013 information technology — security techniques — anonymous digital signatures — part 2: Mechanisms using a group public key., 2013.

ISO16. ISO/IEC. Information technology — security techniques — blind digital signatures — part 2: Discrete logarithm based mechanisms, 2016.

JRS22. C. Jeudy, A. Roux-Langlois, and O. Sanders. Lattice-based signature with efficient protocols, revisited. *IACR Cryptol. ePrint Arch.*, page 509, 2022.

LLM+16. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT*, 2016.

LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *IACR Cryptol. ePrint Arch.*, page 284, 2022.

LNPS21. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *ASIACRYPT*, 2021.

LS15. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 2015.

LW15. V. Lyubashevsky and D. Wichs. Simple lattice trapdoor sampling from a broad class of distributions. In *Public Key Cryptography*, 2015.

LW20. F.-H. Liu and Z. Wang. Rounding in the rings. In *CRYPTO*, 2020.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, 2012.

MP12. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

MP13. D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO*, 2013.

MR07. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 2007.

NIS. NIST. Post-quantum cryptography standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

Pei10. C. Peikert. An efficient and parallel gaussian sampler for lattices. In *CRYPTO*, 2010.

PFH+20. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *FALCON. Tech. rep.*, 2020. Available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

PR06. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, 2006.

RS13. M. Rückert and D. Schröder. Aggregate and verifiably encrypted signatures from multilinear maps without random oracles. *IACR Cryptol. ePrint Arch.*, page 20, 2013.

ZY22. Shiduo Zhang and Yang Yu. Towards a simpler lattice gadget toolkit. In *PKC*, 2022.