Efficient Lattice-Based Blind Signatures via Gaussian One-Time Signatures

Vadim Lyubashevsky[1], Ngoc Khanh Nguyen[1,2], and Maxime Plancon[1,2]

[1] IBM Research – Zurich, Switzerland
[2] ETH Zurich, Switzerland

Abstract. Lattice-based blind signature schemes have been receiving some recent attention lately. Earlier efficient 3-round schemes (Asiacrypt 2010, Financial Cryptography 2020) were recently shown to have mistakes in their proofs, and fixing them turned out to be extremely inefficient and limited the number of signatures that a signer could send to less than a dozen (Crypto 2020). In this work we propose a round-optimal, 2-round lattice-based blind signature scheme which produces signatures of length 150KB. The running time of the signing protocol is linear in the maximum number signatures that can be given out, and this limits the number of signatures that can be signed per public key. Nevertheless, the scheme is still quite efficient when the number of signatures is limited to a few dozen thousand, and appears to currently be the most efficient lattice-based candidate.

Keywords: Lattice Cryptography, Blind Signatures

1 Introduction

Recent years have seen an influx of efficient lattice-based constructions of various cryptographic primitives. From zero-knowledge proofs [BLS19, YAZ+19, ESLL19, LNS20], to group signatures [dPLS18], and even Monero-like private payment systems [EZS+19, LNS21b], it now appears that a lot of fairly advanced privacy-enhancing constructions can be instantiated based on the potential quantum-safety of lattice problems. Somewhat surprisingly, though, there aren't any practical proposals of blind signatures.

Blind signatures, originally proposed by Chaum [Cha82] consist of an interactive procedure between a user and a signer in which the user would like to obtain the signature of a message μ under the public key of the signer, but not reveal the μ to the signer. Furthermore, after producing some certificate that he indeed has a signature of μ, the signer should not be able to figure out during which interaction this certificate was obtained. And of course, it is also required that the user cannot produce signatures by himself – that is, after interacting k times with the signer, the user should not be able to produce $k+1$ signatures.

A candidate for a 3-round lattice-based blind signature has been proposed by Ruckert [Rüc10], and then improved upon in [ABB20]. The proofs of these schemes have, however, recently been shown to be incorrect [HKLN20]. At a high level, the difficulty that was incorrectly overcome was what prevented

Pointcheval and Stern [PS00] from giving stronger proofs for Schnorr's blind signature. It recently turned out that the obstacle blocking the proof was real, and the full Schnorr blind signature has been completely broken [BLL+21]. It is thus quite possible that the errors in [Rüc10, ABB20] are not just mistakes in the proof.

There have since been other constructions of blind signatures, such as [HKLN20], which result in signatures being several (dozen) megabytes long and, more importantly, only allow one to securely sign less than a dozen messages per public key. Another recent proposal [ASY21] produces signatures that are almost as short as in regular lattice-based signatures (i.e. a few kilobytes); but the scheme has a few major downsides. The idea behind the scheme is for the user to encrypt his message μ, and then for the signer to run a (modified version of) the Dilithium lattice-based signature scheme [DKL+18] *homomorphically* by employing a fully-homomorphic encryption scheme. The user would then decrypt and reveal the signature. This approach entails evaluating cryptographic hash functions homomorphically. Furthermore, as it is, the scheme is only blind with respect to an honest signer. To protect against a malicious signer, the signer would be required to give a zero-knowledge proof that the homomorphic evaluation of the signing procedure was done correctly. The extremely heavy tools required for communication between the user and signer almost certainly put this scheme into the theoretical category.

1.1 Our results

We propose a practical two-round lattice-based blind signature scheme with two restrictions. The first is that the signer is required to keep a counter as a state. Secondly, the running time of signature generation and verification is linear in the total number of signatures allowed by the scheme, and so it seems reasonable to put a limit of the total number of signatures to somewhere under 2^{20}.[3]

The signature size is around 150KB, and the interaction between the user and the signer is approximately 16MB. The size of the public key is a little over a megabyte. The 150KB signatures are about 50X longer than the signature size of regular lattice signatures (e.g. [DKL+18, PFH+17], but as far as we're aware these are the shortest (instantiable and having a security proof) blind signatures which are potentially quantum-safe. Even though the communication between the user and the signer is large, all operations are efficient operations on polynomials which have been extensively optimized in recent works on lattice cryptography, and so time-wise, it should be rather efficient. We should mention that the running time of the interaction between the user and the signer is independent of the total number of signatures, and it's only the user's offline time after interacting with the signer that is linear in the total number of signatures.

[3] If one is content with a relaxed definition of blindness where a signature is hidden among T user-signer interactions, then the running time of the scheme can be kept to $O(T)$. This is not a standard definition of a blind signature, but we just mention this possibility in case it's good enough for an application.

A part of our construction requires the use of lattice-based one-time signatures, and we employ ideas from the scheme in [LM18]. In the current paper, we need to use Gaussian-generated secret keys (because in the scheme, they will be sampled using a trapdoor, and the most efficient such algorithm produces Gaussians) unlike the uniform ones used in that paper, and so we develop a different, and arguably easier, security proof for the one-time signature scheme. The developed techniques for analyzing the security of the one-time signature are then extended to prove security of our blind signature and we believe that they can sometimes be used in lieu of analysis that employs Renyi techniques. We believe that this contribution could be potentially of interest in other works.

1.2 Scheme Overview

Let N be the maximum number of messages that can be signed. For each N, we will create a public key and secret key pair for a one-time signature scheme. The N public key pairs are polynomial vectors $(\mathbf{v}_i, \mathbf{w}_i)$, and the corresponding secret keys are polynomial vectors $(\mathbf{s}_i, \mathbf{y}_i)$ with small coefficients satisfying $\mathbf{A}\mathbf{s}_i = \mathbf{v}_i$ and $\mathbf{A}\mathbf{y}_i = \mathbf{w}_i$. All the polynomials are in the polynomial ring $\mathbb{Z}_q[X]/(X^d + 1)$. The matrix $\mathbf{A}$, which is also part of the public key, is generated by the signer together with a trapdoor which allows him to produce the aforementioned short polynomial vectors $\mathbf{s}_i$ and $\mathbf{y}_i$. The public keys $(\mathbf{v}_i, \mathbf{w}_i)$ are uniformly-random and therefore do not need to be stored, as they can simply be defined as $\mathcal{H}(i) = (\mathbf{v}_i, \mathbf{w}_i)$, where $\mathcal{H}$ is some cryptographic hash function such as SHAKE. Thus the public key size is dominated by $\mathbf{A}$ and is not dependent on N.

The message μ is a polynomial with very small, i.e. -1/0/1 coefficients, and the signing process begins by the user sending an encryption $\mathbf{c} = \mathsf{enc}(\mu)$. The signer's goal is to apply a function f to $\mathbf{c}$ such that $\mathsf{dec}(f(\mathbf{c})) = \mathbf{s}_i\mu + \mathbf{y}_i$. The signer thus sends $f(\mathbf{c})$ to the user, and the latter obtains $\mathbf{z} = \mathbf{s}_i\mu + \mathbf{y}_i$ by applying dec. The vector $\mathbf{z}$ has small coefficients and satisfies the relation

$$\mathbf{A}\mathbf{z} = \mathbf{v}_i\mu + \mathbf{w}_i. \tag{1}$$

The vector $\mathbf{z}$ is a signature of μ, but the user cannot reveal it in the clear because that would allow the signer to link the message to the instance during which it was signed. Instead, the user outputs a zero-knowledge proof of knowledge of a $\mathbf{z}$ with small coefficients satisfying (1) for some $(\mathbf{v}_i, \mathbf{w}_i)$ from a set. Such a compact proof, whose size is logarithmic in N, was given in [LNS21b]. Since this proof does not reveal the $\mathbf{z}$ nor the specific $(\mathbf{v}_i, \mathbf{w}_i)$ from the set of public keys, the blindness property is preserved.

The main technical part of this work is showing that for our specific functions $\mathsf{enc}, \mathsf{dec}$, and f, the message μ is hidden, and that $f(\mathbf{c})$ does not leak enough information about the signer's keys $\mathbf{s}_i, \mathbf{y}_i$. In particular, the user who obtains $f(\mathbf{c})$ should not be able to produce two different $(\mathbf{z}, \mu), (\mathbf{z}', \mu')$ satisfying (1).

An easy solution for hiding the μ and not having $f(\mathbf{c})$ leak anything would be to use a circuit-private homomorphic encryption scheme; but this would be overkill. We instead show a solution, which is similar in intuition to the one-time

signature proof idea in [LM18], which does not require the secrets $\mathbf{s}_i$ and $\mathbf{y}_i$ to be completely hidden either by drowning them with noise or applying a Renyi entropy argument. Instead, it's enough to show that $\mathbf{z}$ does not leak the exact $\mathbf{s}_i$ and $\mathbf{y}_i$. And in this case, coming up with another signature is as hard as solving the SIS problem.

We now give some more details. To improve readability, we will drop the subscripts i from the secret and public keys. Because all the keys are independent, we can prove things about individual public/secret key pairs. The $\mathsf{enc}(\mu)$ procedure is essentially an LWE public key encryption scheme in which the user both does the encrypting and decrypting. So the public key consists of a random matrix $\mathbf{B}$ and a polynomial vector $\mathbf{b}^T = \mathbf{x}^T\mathbf{B}$, where $\mathbf{x}$ is a polynomial vector with small coefficients. The $\mathsf{enc}(\mu)$ function samples random small-coefficient polynomial vectors $\mathbf{r}, \mathbf{e}$ and a polynomial e', and outputs the ciphertext $(\mathbf{t}, t') = (\mathbf{Br} + p\mathbf{e}, \mathbf{b}^T\mathbf{r} + pe' + \mu)$, where p is a "large-enough" prime. This ciphertext, along with a zero-knowledge proof that it was properly computed (i.e. that $\mathbf{r}, \mathbf{e}, e', \mu$ have small coefficients) is sent to the signer. The zero-knowledge proof can be created using the fairly-efficient recent techniques from [ALS20, ENS20, LNS21a].

The signer now needs to create an encryption of $\mathbf{z} = \mathbf{s}\mu + \mathbf{y}$. He does this by creating an encryption of each coefficient comprising $\mathbf{z}$ independently. In particular, if $\mathbf{s} = \begin{bmatrix} s_1 \\ \dots \\ s_\alpha \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ \dots \\ y_\alpha \end{bmatrix}$, then for each $1 \leqslant j \leqslant \alpha$, the signer computes

$$s_j\mathbf{t} = \mathbf{Br}s_j + p\mathbf{e}s_j \tag{2}$$

$$y_j + s_j t' = \mathbf{b}^T\mathbf{r}s_j + pe's_j + (\mu s_j + y_j). \tag{3}$$

Because $\mathbf{r}, \mathbf{e}, e'$, and s_j have small coefficients, and assuming that all the coefficients of $(\mu s_j + y_j)$ are less than p, the above is an encryption of $(\mu s_j + y_j)$. That is, one would decrypt in the usual way by computing

$$(y_j + s_j t') - \mathbf{x}^T s_j \mathbf{t} \bmod p = \mu s_j + y_j.$$

It's unclear however, whether sending (2) and (3) is secure on the signer's part. That is, he is possibly leaking too much information about $\mathbf{s}$ and $\mathbf{y}$. Instead of (2) and (3), he therefore sends the "masked" equations

$$s_j\mathbf{t} + \mathbf{B}\mathbf{y}'_j + p\mathbf{y}''_j = \mathbf{B}(\mathbf{r}s_j + \mathbf{y}'_j) + p(\mathbf{e}s_j + \mathbf{y}''_j) \tag{4}$$

$$y_j + s_j t' + \mathbf{b}^T\mathbf{y}'_j + py''' = \mathbf{b}^T(\mathbf{r}s_j + \mathbf{y}'_j) + p(e's_j + y'''_j) + (\mu s_j + y_j), \tag{5}$$

where $\mathbf{y}'_j, \mathbf{y}''_j$, and y'''_j are (vectors of) polynomials with small coefficients. As long as these coefficients are small enough, one should still be able to decrypt $\mu s_j + y_j$ as before. We will now outline the proof showing that an adversary who is able to produce a signature other than $\mu\mathbf{s} + \mathbf{y}$ for the public key $(\mathbf{v}, \mathbf{w})$ and message μ can solve the SIS problem.

In the real scheme, the public key is set as $(\mathbf{v}, \mathbf{w}) = \mathcal{H}(i)$, and then the $\mathbf{s}, \mathbf{y}$ are sampled using a trapdoor for $\mathbf{A}$. In the security proof, we will instead get a random $\mathbf{A}$ from the challenger, sample the $\mathbf{s}, \mathbf{y}$, and then program the random oracle $\mathcal{H}(i) = (\mathbf{v} = \mathbf{As}, \mathbf{w} = \mathbf{Ay})$. Because the trap-doored matrix is indistinguishable from uniform [MP12] and the standard deviation of $\mathbf{s}, \mathbf{y}$ is above the smoothing parameter [MR07], the two distributions are indistinguishable. The reduction's goal is to now solve SIS for the matrix $\mathbf{A}$.

Because the reduction knows the secret keys $\mathbf{s}, \mathbf{y}$, it can produce the responses in (4) and (5). Now, suppose that an adversary who sees (4) and (5) is able to create two valid signatures $\mathbf{z}, \mathbf{z}'$ (with small coefficients) for two messages $\mu \neq \mu'$ satisfying[4]

$$\mathbf{Az} = \mathbf{v}\mu + \mathbf{w} \tag{6}$$

$$\mathbf{Az}' = \mathbf{v}\mu' + \mathbf{w}. \tag{7}$$

Plugging in $(\mathbf{v} = \mathbf{As}, \mathbf{w} = \mathbf{Ay})$ and subtracting, the reduction obtains

$$\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{A}((\mathbf{s}(\mu - \mu')). \tag{8}$$

Thus, if

$$\mathbf{z} - \mathbf{z}' \neq \mathbf{s}(\mu - \mu'), \tag{9}$$

the reduction extracted a solution to SIS. The crucial part is now proving that the signatures produced by the forger will indeed satisfy this inequality with some non-negligible probability. Notice that if $\mathbf{z} - \mathbf{z}' = \mathbf{s}(\mu - \mu')$, then one has also has extracted $\mathbf{s}$ (because the coefficients of $\mathbf{s}(\mu - \mu')$ are small-enough that no reduction modulo q takes place and so the ring $\mathbb{Z}[X]/(X^d + 1)$ is an integral domain, and so one can simply divide by $\mu - \mu'$). In other words, the reduction can either extract a solution to SIS from the adversary, or the adversary "knows" the value $\mathbf{s}$ that was used by the reduction. The former is the computational assumption upon which the scheme is based, while the latter, we will show, is information-theoretically impossible except with probability at most $1 - \delta$. It's important to point out that the latter holds *for all* views that contain the public key and equations (4) and (5). Therefore, it is impossible for an adversary to always extract the correct $\mathbf{s}$, and so (9) will be satisfied with probability at least δ. So if an adversary succeeds in a forgery with probability ϵ,the reduction will solve SIS with probability $\epsilon\delta$.

We now need to show that that despite knowing the public keys and having access to (4) and (5), the adversary still cannot *information-theoretically* determine the exact value $\mathbf{s}$. Consider the possibility that instead of the vector $\mathbf{s}$, we sampled the vector $\tilde{\mathbf{s}} = \mathbf{s} + \mathbf{u}$, where $\mathbf{u}$ satisfies $\mathbf{Au} = \mathbf{0}$. This is a valid pre-image for the public key $\mathbf{v} = \mathbf{As} = \mathbf{A}(\mathbf{s} + \mathbf{u})$, and in order to also satisfy (4),(5), we

[4] The first signature $\mathbf{z}$ on a message μ is already given to the adversary in (4) and (5), so he really just has to produce a second one

would need to have sampled, instead of $\mathbf{y}, \mathbf{y}', \mathbf{y}''$, and y''',

$$\tilde{\mathbf{y}} = \mathbf{y} - \mu\mathbf{u} \tag{10}$$

$$\tilde{\mathbf{y}}'_j = \mathbf{y}'_j - u_j\mathbf{r} \tag{11}$$

$$\tilde{\mathbf{y}}''_j = \mathbf{y}''_j - u_j\mathbf{e} \tag{12}$$

$$\tilde{y}'''_j = y'''_j - u_j e'. \tag{13}$$

To complete the proof, we need to show that the event that $\mathbf{s}, \mathbf{y}, \mathbf{y}'_j, \mathbf{y}''_j, y'''_j$ are sampled, conditioned on the view of the adversary, is not overly dominant. For simplicity, let's just look at $\mathbf{s}$ and the alternative $\tilde{\mathbf{s}} = \mathbf{s} + \mathbf{u}$; incorporating the $\tilde{\mathbf{y}}, \tilde{\mathbf{y}}', \tilde{\mathbf{y}}'', \tilde{y}'''$ in the analysis is done in a similar manner.

If $\mathbf{s}$ is sampled from $\mathbb{Z}^m$ according to a Gaussian distribution with standard deviation σ – that is the distribution is proportional to $\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \cdot \rho_\sigma(\mathbf{x})$, where $\rho_\sigma(\mathbf{x}) = e^{-\|\mathbf{x}\|^2/2\sigma^2}$, then the conditional probability that $\mathbf{s}$ is some $\mathbf{s}^*$ satisfying $\mathbf{v} = \mathbf{A}\mathbf{s}^*$ is

$$\Pr_{\mathbf{s}*}[\mathbf{s}* = \mathbf{s} | \mathbf{A}\mathbf{s} = \mathbf{v}] = \frac{\rho_\sigma(\mathbf{s})}{\sum\limits_{\mathbf{u}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{s}-\mathbf{u})} \leqslant \frac{1}{\sum\limits_{\mathbf{u}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{u})}, \tag{14}$$

where the last inequality is implicit in the proof of [AR04, Lemma 3.2]. In Theorem 3.3, we then show that when $\sigma \approx q^{n/m}$, the above inequality is less than $\frac{1}{2}$, and so even an all-powerful adversary cannot know the exact $\mathbf{s}^*$.

In Section 3, as an interlude, we also use the same techniques to give an instantiation of the one-time signature from [LM18] where the secret keys are Gaussians. In particular, the one-time signatures are just the blind signatures without the blinding part and without the user needing to hide the public key that was used to sign the message. That is, there is no user and no equations (4) and (5). The signer simply sends $\mathbf{z} = \mathbf{s}\mu + \mathbf{y}$ as his signature of μ, and the verifier checks that $\|\mathbf{z}\|$ is small and $\mathbf{A}\mathbf{z} = \mathbf{v}\mu + \mathbf{w}$. This is exactly the template from [LM18], but with a different security proof which crucially uses the fact that the secret keys are Gaussian instead of uniform. It seems that both instantiations are about equally efficient, but we include this instantiation in case a Gaussian-based scheme is useful for some application, similarly to how it was extended in this paper.

As a side note, we would like to draw attention to the advantage of our proof over a more "generic" one that would use Renyi entropy arguments (e.g. [BLL+15]) to show that not enough information about $\mathbf{s}$ is leaked in (4) and (5)). Using such arguments would require to set the standard deviations of $\mathbf{y}, \mathbf{y}'$, etc. to be at least as large as $\|\mathbf{s}\mu\|, \|\mathbf{r}s_j\|$, etc. Our proof technique, on the other hand, only needs the standard deviation to be approximately $q^{n/m}$, which is siginficantly smaller because just $\mathbf{s}$ has standard deviation at least that. In fact, somewhat counter-intuitively, one does not even need the "mask" $\mathbf{y}$ to have larger standard deviation than $\|\mathbf{s}\mu\|$. This is a rather different situation than in signature schemes where the role of $\mathbf{y}$ is to make the distribution of $\mathbf{y} + \mathbf{s}\mu$ independent of $\mathbf{s}$.

We remark, however, that our technique cannot replace the Renyi argument everywhere. For our technique to be applicable, the reduction needs to know the secret values when performing the simulation, because we do do not make any claims about what the output distribution looks like. Renyi proofs, on the other hand, argue that the resulting distribution is "close-enough" to some distribution which can be sampled without knowing the secret.

2 Preliminaries

2.1 Notation

Let q be an odd prime and λ be a security parameter. In this paper we aim for 128-bit security. Unless stated otherwise, all algorithms are implicitly given a security parameter in unary. The joint execution of two algorithms $\mathcal{A}$ and $\mathcal{B}$ in an interactive protocol with private inputs x to $\mathcal{A}$ and y to $\mathcal{B}$ is written as $(a, b) \leftarrow \langle \mathcal{A}(x), \mathcal{B}(y) \rangle$ where a and b are the private outputs of $\mathcal{A}$ and $\mathcal{B}$ respectively.

We write $x \leftarrow S$ when $x \in S$ is sampled uniformly at random from the finite set S and similarly $x \leftarrow D$ when x is sampled according to the discrete distribution D. The statistical distance between two probability distributions X and Y over a countable set D is defined as $\Delta(X, Y) = \sum_{d \in D} |X(d) - Y(d)|$. For integer $n \in \mathbb{N}$, we define $[n] := \{1, 2, \ldots, n\}$. Given two functions $f, g : \mathbb{N} \to [0, 1]$, we write $f(\mu) \approx g(\mu)$ if $|f(\mu) - g(\mu)| < \mu^{-\omega(1)}$. A function f is negligible if $f \approx 0$. We write $\mathsf{negl}(n)$ to denote an unspecified negligible function in n.

2.2 Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ consist of n linearly independent vectors. The n-dimensional lattice generated by B is defined as

$$\mathbf{\Lambda} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} c_i \mathbf{b}_i : c_1, \ldots, c_n \in \mathbb{Z} \right\}.$$

The dual lattice of $\mathbf{\Lambda}$ is defined as $\mathbf{\Lambda}^* = \{\mathbf{x} \in \mathbb{R}^n : \forall \mathbf{v} \in \mathbf{\Lambda}, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}$. We denote $\tilde{\mathbf{B}}$ to be the Gram-Schmidt orthogonalization of $\mathbf{B}$.

For a power of two d, denote $\mathcal{R}$ and $\mathcal{R}_q$ respectively to be the rings $\mathbb{Z}[X]/(X^d + 1)$ and $\mathbb{Z}_q[X]/(X^d + 1)$. Unless stated otherwise, lower-case letters denote elements in $\mathcal{R}$ or $\mathcal{R}_q$ and bold lower-case (resp. upper-case) letters represent column vectors (resp. matrices) with coefficients in $\mathcal{R}$ or $\mathcal{R}_q$.

For an element $w \in \mathbb{Z}_q$, we write $\|w\|_\infty$ to mean $|w \bmod^{\pm} q|$. Define the ℓ_∞ and ℓ_p norms for $w = w_0 + w_1 X + \ldots + w_{d-1} X^{d-1} \in \mathcal{R}$ as follows:

$$\|w\|_\infty = \max_j \|w_j\|_\infty, \quad \|w\|_p = \sqrt[p]{\|w_0\|_\infty^p + \ldots + \|w_{d-1}\|_\infty^p}.$$

If $w = (w_1, \ldots, w_m) \in \mathcal{R}^k$, then

$$\|\mathbf{w}\|_\infty = \max_j \|w_j\|_\infty, \quad \|\mathbf{w}\|_p = \sqrt[p]{\|w_1\|^p + \ldots + \|w_k\|^p}.$$

By default, $\|\mathbf{w}\| := \|\mathbf{w}\|_2$. Similarly, we define the norms for vectors over $\mathbb{Z}_q$. Denote $S_\gamma = \{x \in \mathcal{R}_q : \|x\|_\infty \leqslant \gamma\}$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we define the module q-ary lattice as:

$$\mathbf{\Lambda}_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}.$$

Similarly, when $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ then:

$$\mathbf{\Lambda}_{\mathcal{R}_q}^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathcal{R}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \text{ over } \mathcal{R}_q\}.$$

For a polynomial $f = f_0 + f_1 X + \ldots + f_{d-1} X^{d-1} \in \mathcal{R}$, we define the rotation matrix $\mathsf{Rot}(f) \in \mathbb{Z}^{d \times d}$ as:

$$\mathsf{Rot}(f) = \begin{bmatrix} f_0 & -f_{d-1} & \ldots & -f_1 \\ f_1 & f_0 & \ldots & -f_2 \\ \vdots & \vdots & \ldots & \vdots \\ f_{d-1} & f_{d-2} & \cdots & f_0 \end{bmatrix}.$$

Similarly, for a matrix $\mathbf{F} = (f_{i,j}) \in \mathcal{R}_q^{n \times m}$, we define

$$\mathsf{Rot}(\mathbf{F}) = \begin{bmatrix} \mathsf{Rot}(f_{1,1}) & \mathsf{Rot}(f_{1,2}) & \ldots & \mathsf{Rot}(f_{1,m}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathsf{Rot}(f_{n,1}) & \mathsf{Rot}(f_{n,2}) & \ldots & \mathsf{Rot}(f_{n,m}) \end{bmatrix} \in \mathbb{Z}^{nd \times md}.$$

2.3 Discrete Gaussian Distribution on Lattices

For any $\sigma > 0$, we define the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c} \in \mathbb{R}^n$ with parameter σ as:

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{\sigma,\mathbf{c}}(\mathbf{x}) := \exp\left(-\|\mathbf{x} - \mathbf{c}\|^2 / 2\sigma^2\right).$$

More generally, if $\sigma = (\sigma_1, \ldots, \sigma^n) \in \mathbb{R}_{>0}^n$ then we define $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \prod_{i=1}^n \rho_{\sigma_i,c_i}(x_i)$[5]. When we omit the subscript $\mathbf{c}$, we set $\mathbf{c} = \mathbf{0}$ by default.

Let $\mathbf{c} \in \mathbb{R}^n, \sigma > 0$ and $\mathbf{\Lambda}$ be a n-dimensional lattice. We now define the discrete Gaussian distribution over a lattice $\mathbf{\Lambda}$ as

$$\forall \mathbf{x} \in \mathbf{\Lambda}, D_{\mathbf{\Lambda},\sigma,\mathbf{c}}(\mathbf{x}) := \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\mathbf{\Lambda})}.$$

As above, we may omit the subscript $\mathbf{c}$. Also, we drop the subscript $\mathbf{\Lambda}$ when $\mathbf{\Lambda} = \mathbb{Z}^n$ and denote it as $D_{\sigma,\mathbf{c}}^n$.

We recall the definition of a smoothing parameter [MR07].

[5] One could define the Gaussian function more generally using a covariance matrix. However, we will not need such a general case and thus we omit it for presentation purposes.

Definition 2.1. *Let $\mathbf{\Lambda}$ be an n-dimensional lattice and $\varepsilon > 0$. Then, the smoothing parameter $\eta_\varepsilon(\mathbf{\Lambda})$ is the smallest real $s > 0$ such that $\rho_{\frac{1}{\sqrt{2\pi}s}}(\mathbf{\Lambda}^* \backslash \{\mathbf{0}\}) \leqslant \varepsilon$.*

We will use the following upper-bound on the smoothing parameter.

Lemma 2.2 ([GPV08]). *For any n-dimensional lattice $\mathbf{\Lambda}$ with basis $\mathbf{B}$ and $\varepsilon > 0$, we have:*

$$\eta_\varepsilon(\mathbf{\Lambda}) \leqslant \|\tilde{\mathbf{B}}\| \cdot \sqrt{\ln(2n/(1+1/\varepsilon))/\pi}.$$

The next fact states that the total Gaussian measure on any translation of the lattice is essentially the same.

Lemma 2.3 ([MR07]). *Let $\mathbf{\Lambda}$ be an n-dimensional lattice. Then, for any $\varepsilon \in (0,1)$, $\sigma \geqslant \eta_\varepsilon(\mathbf{\Lambda})$ and $\mathbf{c} \in \mathbb{R}^n$, we have*

$$\rho_{\sigma,\mathbf{c}}(\mathbf{\Lambda}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot \rho_\sigma(\mathbf{\Lambda}).$$

In this paper we will apply the following simple corollary and provide the proof in Appendix A.

Corollary 2.4. *Let $\mathbf{\Lambda}, \mathbf{\Lambda}'$ be n-dimensional lattices and $\mathbf{\Lambda}' \subseteq \mathbf{\Lambda}$. Then, for any $\varepsilon \in (0, \frac{1}{2}]$, $\sigma \geqslant \eta_\varepsilon(\mathbf{\Lambda}')$, define the following probability distributions D_1, D_2:*

- *D_1: first sample $\mathbf{x} \leftarrow D_{\mathbf{\Lambda},\sigma}$ and output $(\mathbf{x}, \mathbf{t} := \mathbf{x} \bmod \mathbf{\Lambda}')$,*
- *D_2: first generate $\mathbf{t}$ uniformly at random from $\mathbf{\Lambda} \backslash \mathbf{\Lambda}'$ and then sample $\mathbf{x} \leftarrow D_{\mathbf{\Lambda},\sigma}$ conditioned on $\mathbf{t} = \mathbf{x} \bmod \mathbf{\Lambda}'$. Output $(\mathbf{x}, \mathbf{t})$.*

Then, $\Delta(D_1, D_2) \leqslant 4\varepsilon$.

We will use the following tail bound from [Ban93, Lyu12].

Lemma 2.5. *Let $m, k > 1, \mathbf{\Lambda}$ be m-dimensional lattice and $\mathbf{c} \in \mathbb{Z}^m$. Then*

1. $\Pr_{z \leftarrow D_\sigma}\left[|z| > k\sigma\right] \leqslant 2e^{\frac{-k^2}{2}}$.
2. $\Pr_{\mathbf{z} \leftarrow D_\sigma^m}\left[\|\mathbf{z}\|_2 > k\sigma\sqrt{m}\right] \leqslant k^m e^{\frac{m}{2}(1-k^2)}$.
3. $\Pr_{\mathbf{z} \leftarrow D_{\mathbf{\Lambda},\sigma,\mathbf{c}}^m}\left[\|\mathbf{z}\|_2 > k\sigma\sqrt{m}\right] \leqslant 2k^m e^{\frac{m}{2}(1-k^2)}$.

2.4 Module-SIS and Module-LWE Problems

Security of our blind signature scheme relies on the well-known computational lattice problems, namely Module-LWE (MLWE) and Module-SIS (MSIS) [LS15]. Both problems are defined over $\mathcal{R}_q$.

Definition 2.6 ($\mathsf{MSIS}_{n,m,B}$). *Given $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, the* Module-SIS *problem with parameters $n, m > 0$ and $0 < B < q$ asks to find $\mathbf{z} \in \mathcal{R}_q^m$ such that $\mathbf{Az} = \mathbf{0}$ over $\mathcal{R}_q$ and $0 < \|\mathbf{z}\| \leqslant B$. An algorithm $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MSIS}_{n,m,B}$ if*

$$\Pr\left[0 < \|\mathbf{z}\| \leqslant B \ \wedge \ \mathbf{Az} = \mathbf{0} \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}; \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A})\right] \geqslant \epsilon.$$

Definition 2.7 ($\mathsf{MLWE}_{n,m,\chi}$). *The* Module-LWE *problem with parameters $n, m > 0$ and an error distribution χ over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases: 1) $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ for $\mathbf{A} \leftarrow \mathcal{R}_q^{n\times m}$, a secret vector $\mathbf{s} \leftarrow \chi^m$ and error vector $\mathbf{e} \leftarrow \chi^n$, and 2) $(\mathbf{A}, \mathbf{b}) \leftarrow \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^m$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MLWE}_{n,m,\chi}$ if*

$$\begin{aligned}\Big|\Pr\left[b=1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{n\times m};\, \mathbf{s} \leftarrow \chi^m;\, \mathbf{e} \leftarrow \chi^n;\, b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{As}+\mathbf{e})\right] \\ - \Pr\left[b=1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{n\times m};\, \mathbf{b} \leftarrow \mathcal{R}_q^{n};\, b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b})\right]\Big| \geqslant \epsilon.\end{aligned} \tag{15}$$

2.5 Blind Signatures

We present a definition of a blind signature where the signer is stateful and a user is allowed to make at most $k = \mathsf{poly}(\lambda)$ queries.

Definition 2.8. *A k-time stateful blind signature scheme* BS *consists of PPT algorithms* $\mathsf{BS.KeyGen}$, $\mathsf{BS.Ver}$ *along with two interactive PPT algorithms $\mathcal{S}$ and $\mathcal{U}$ such that*

- $\mathsf{BS.KeyGen}(1^\lambda, 1^k)$*: given a security parameter λ and maximum number of signing queries k, it outputs a private/verification key pair* $(\mathsf{sk}, \mathsf{pk})$,
- *For $i \in [k]$, the joint execution of $\mathcal{S}(\mathsf{sk}, i)$ and $\mathcal{U}(\mathsf{pk}, m)$, where $m \in \{0,1\}^*$, generates an output σ_i for the user $\mathcal{U}$ and no output for $\mathcal{S}$, i.e.*

$$(\perp, \sigma) \leftarrow \langle \mathcal{S}(\mathsf{sk}, i), \mathcal{U}(\mathsf{pk}, m) \rangle$$

- $\mathsf{BS.Ver}(\mathsf{pk}, m, \sigma)$*: given a verification key* pk*, message m and signature σ, it outputs a bit b.*

The main difference from previous works is the fact that $\mathcal{S}$ has the additional input i which can be seen as a state. Indeed, if the message m and random coins in the system are fixed, then it might still be the case that for $i \neq j$, the interaction between $\mathcal{S}(\mathsf{sk}, i)$ and $\mathcal{U}(\mathsf{pk}, m)$ would be different than the interaction between $\mathcal{S}(\mathsf{sk}, j)$ and $\mathcal{U}(\mathsf{pk}, m)$.

In general, blind signatures must satisfy three properties: (i) correctness, (ii) blindness and (iii) one-more unforgeability. We adapt these standard properties to k-time stateful blind signatures in an intuitive way.

Definition 2.9 (Correctness). *A k-time stateful blind signature scheme* BS *is correct if for any k messages $m_1, \ldots, m_k \in \{0,1\}^*$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KeyGen}(1^\lambda)$, and σ_i output by $\mathcal{U}$ in the joint execution between $\mathcal{S}(\mathsf{sk}, i)$ and $\mathcal{U}(\mathsf{pk}, m_i)$ for $i \in [k]$, it holds that $\forall i \in [k], \mathsf{BS.Ver}(\mathsf{pk}, m_i, \sigma_i) = 1$ with probability $1 - \mathsf{negl}(\lambda)$.*

Definition 2.10 (Blindness). *A k-time stateful blind signature scheme* BS *is blind every PPT algorithm $\mathcal{S}^*$ wins the following blindness game with negligible probability:*

1. $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{S}^*$.
2. *$\mathcal{S}^*$ provides two distinct messages m_0, m_1.*

3. $b \leftarrow \{0,1\}$.
4. $\mathcal{S}^*$ *interacts concurrently with* $\mathcal{U}_0 = \mathcal{U}(\mathsf{pk}, m_b)$ *and* $\mathcal{U}_1 = \mathcal{U}(\mathsf{pk}, m_{1-b})$.
5. *If either* $\mathcal{U}_0$ *or* $\mathcal{U}_1$ *abort, then* $(\sigma_0, \sigma_1) = (\perp, \perp)$. *Otherwise, denote* σ_b *and* σ_{1-b} *to be the outputs of* $\mathcal{U}_0$ *and* $\mathcal{U}_1$ *respectively. Then,* $\mathcal{S}^*$ *is given* (σ_0, σ_1).
6. $\mathcal{S}^*$ *returns a bit* b'. *It wins the blindness game if* $b = b'$.

In this paper we consider blindness in the malicious signer model i.e. an adversary gets to choose its own keys.

Definition 2.11 (One-More Unforgeability). *A k-time stateful blind signature scheme* BS *is one-more unforgeable if every PPT algorithm* $\mathcal{U}^*$ *wins the following one-more unforgeability game with negligible probability:*

1. $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KeyGen}(1^\lambda)$ *and* $\mathcal{U}^*$ *is given* pk.
2. $\mathcal{U}^*$ *interacts with* ℓ *signers* $\mathcal{S}(\mathsf{sk}, 1), \ldots, \mathcal{S}(\mathsf{sk}, \ell)$ *where* $\ell \leqslant k$.
3. $\mathcal{U}^*$ *outputs* $\ell + 1$ *pairs* (m_i, σ_i) *where* $i \in [\ell + 1]$.
4. *Algorithm* $\mathcal{U}^*$ *wins the one-more unforgeability game if* $\forall i \in [\ell + 1]$, *it holds that* $\mathsf{BS.Ver}(\mathsf{pk}, m_i, \sigma_i) = 1$.

2.6 Lattice-Based NIZKs

We will use the LANES framework for efficient (non-interactive) arguments of knowledge for proving linear and multiplicative relations between committed messages developed in [ALS20, ENS20, LNS21a][6]. In this paper we are interested in the following two relations.

Verifiable Encryption. We want to prove that a ciphertext was constructed correctly. More concretely, let μ be a binary polynomial, p be prime, $\mathbf{r} \in S_\gamma^m$ and $\mathbf{e} \in S_\gamma^{n+1}$ be randomness and error vectors respectively. Then, given public keys $\mathbf{B} \in \mathcal{R}_q^{n \times m}, \mathbf{b} \in \mathcal{R}_q^m$ and valid ciphertext $\mathbf{t} \in \mathcal{R}_q^{n+1}$, we want to prove that

(i) μ is a binary polynomial,
(ii) $\mathbf{r}$ and $\mathbf{e}$ have coefficients between $-\gamma$ and γ,
(iii) $\begin{pmatrix} \mathbf{B} \\ \mathbf{b}^T \end{pmatrix} \mathbf{r} + p\mathbf{e} + \begin{pmatrix} \mathbf{0} \\ \mu \end{pmatrix} = \mathbf{t}$.

One observes that (i) and (ii) are multiplicative relations and (iii) is a linear relation. These statements can be efficiently proven using protocols from [ENS20, LNS21a]. We will denote the verifiable encryption proof as $\pi_{\mathsf{enc}}((\mathbf{B}, \mathbf{b}, \mathbf{t}), (\mathbf{r}, \mathbf{e}, \mu))$.

One-out-of-many Proof. Our blind signature will consist of the one-out-of-many proof [GK15], i.e. a proof that one of the elements of a public set is a commitment to zero. In our setting, we want to prove that, given a matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ and a public finite set U of vectors in $\mathcal{R}_q^n$, we know a vector $\mathbf{s} \in \mathcal{R}_q^m$

[6] We refer to [ENS20, LNS21a] for more details on the protocol.

which has small coefficients and $\mathbf{As} \in U$. As shown in [GK15], this concept is closely related to ring signatures.

Very recently, Lyubashevsky et al. [LNS21b] proposed an efficient one-out-of-many proof based on the LANES framework where the communication size is logarithmic in the size of U. We will apply the non-interactive protocol from [LNS21b, Section 3.2] and denote this proof as $\pi_{\in}((\mathbf{A}, U), \mathbf{s})$.

3 Lattice Based One-Time Signature Revisited

An important building block of our blind signature is the lattice-based one-time signature construction by Lyubashevsky and Micciancio [LM18] using modules lattices. However, we modify the original scheme so that the secret keys are chosen from a discrete Gaussian distribution rather than picked uniformly at random. The main motivation for such a change is that it mixes well with other building blocks (e.g. trapdoor sampling [MP12]) described in the next section.

The one-time signature is defined by the following algorithms:

- **Key Generation**: sample matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ uniformly at random and a secret key $\mathbf{y} \leftarrow \mathcal{D}_{\sigma_y}^{md}, \mathbf{s} \leftarrow \mathcal{D}_{\sigma_s}^{md}$. Then, the public key is a pair $\mathsf{pk} = (\mathbf{A}, \mathbf{w} := \mathbf{Ay}, \mathbf{v} := \mathbf{As})$ and its corresponding secret key is $\mathsf{sk} = (\mathbf{y}, \mathbf{s})$.
- **Signing**: given a binary polynomial $\mu \in \{0,1\}^d \subset \mathcal{R}_q$ as a message and a secret key $(\mathbf{y}, \mathbf{s})$, it outputs $\mathbf{z} = \mathbf{y} + \mu\mathbf{s}$.
- **Verification**: given a binary polynomial $\mu \in \mathcal{R}_q$, public key $(\mathbf{A}, \mathbf{w}, \mathbf{v})$ and a signature $\mathbf{z}$, it checks whether $\|\mathbf{z}\| \leqslant (\sigma_y + d\sigma_s)\sqrt{2md}$ and $\mathbf{Az} = \mathbf{w} + \mu\mathbf{v}$.

Correctness and security of the one-time signature can be summarised with the following theorem. We provide the full proof in Appendix B.

Theorem 3.1. *Let $m \geqslant 6(\lambda+1)/d$. Then, the one-time signature scheme above is correct. Concretely, the verification algorithm always accepts signatures produced by the legitimate signer with an overwhelming probability.*

For unforgeability, suppose that $\sigma_y \geqslant q^{n/m}\sqrt{2e}d + 2$, $\sigma_s \geqslant q^{n/m}\sqrt{2e} + 2$ and $q > 4d\sigma_s\sqrt{2md}$. If there is an adversary $\mathcal{A}$ which succeeds in breaking the strong unforgeability game of the one-time signature scheme with probability γ, then there exists an algorithm that can solve $\mathsf{MSIS}_{n,m,2(\sigma_y+d\sigma_s)\sqrt{2md}}$ with probability at least $\gamma/3 - \mathsf{negl}(\lambda)$ in essentially the same running time as the forgery attack.

For readability, we first provide a sketch for the unforgeability proof. Namely, assume there is an adversary $\mathcal{A}$ which succeeds in breaking the strong unforgeability game of the one-time signature scheme. We can then construct an algorithm $\mathcal{B}$ for solving MSIS as follows. Given a uniformly random matrix $\mathbf{A}$, the algorithm samples $\mathbf{y} \leftarrow \mathcal{D}_{\sigma_y}^{md}, \mathbf{s} \leftarrow \mathcal{D}_{\sigma_s}^{md}$ and sets $\mathbf{w} = \mathbf{Ay}$ and $\mathbf{v} = \mathbf{As}$. Next, $\mathcal{B}$ outputs $(\mathbf{A}, \mathbf{w}, \mathbf{v})$. When $\mathcal{A}$ asks a signing query on input μ, $\mathcal{B}$ answers with $\mathbf{z} = \mathbf{y} + \mu\mathbf{s}$. Finally, $\mathcal{A}$ outputs a forgery $(\mu', \mathbf{z}')$. Assuming that it is valid, $\mathcal{B}$ outputs a potential solution $\mathbf{z}' - (\mathbf{y} + \mu'\mathbf{s})$. Now, given $\mathbf{A}, \mathbf{w}, \mathbf{v}, \mu, \mathbf{z}$, the adversary $\mathcal{A}$ does not know which $(\mathbf{y}, \mathbf{s})$ from the following set was used:

$$S = \{(\mathbf{y}', \mathbf{s}') : \mathbf{Ay}' = \mathbf{w}, \mathbf{As}' = \mathbf{v}, \mathbf{z} = \mathbf{y}' + \mu\mathbf{s}'\}.$$

To this end, we will prove that the probability of picking $(\mathbf{y}, \mathbf{s})$ when sampling from a discrete Gaussian distribution restricted to S (which is a coset of a lattice as shown below) is sufficiently small, e.g. $1/2$ when standard deviations σ_y, σ_s are chosen properly.

Let us fix $\mathbf{A}, \mathbf{y}, \mathbf{s}, \mu$. We provide tools to compute an upper-bound on the following probability:

$$\Pr\left[\mathbf{y}^* = \mathbf{y} \wedge \mathbf{s}^* = \mathbf{s} : (\mathbf{y}^*, \mathbf{s}^*) \leftarrow D_{\mathbf{\Lambda}^{\perp}_{\mathcal{R}_q}(\mathbf{X}),\sigma,\mathbf{c}}\right] \tag{16}$$

where $\sigma = (\sigma_y, \ldots, \sigma_y, \sigma_s, \ldots, \sigma_s) \in \mathbb{R}^{2m}_{>0}$,

$$\mathbf{X} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \\ 1 \cdot \mathbf{I}_m & \mu \cdot \mathbf{I}_m \end{pmatrix} \in \mathcal{R}_q^{(2n+m)\times 2m} \text{ and } \mathbf{c} = -\begin{pmatrix} \mathbf{y} \\ \mathbf{s} \end{pmatrix} \in \mathcal{R}_q^{2m}.$$

These techniques will be crucial for proving not only unforgeability for the one-time signature but also for one-more unforgeability of the blind signature presented in the next section.

We start with the following technical lemma.

Lemma 3.2. *Let $\mathbf{M} \in \mathbb{Z}^{m\times n}$ and $\mathbf{\Lambda}$ be an n-dimensional lattice. Then, for any $\sigma \in \mathbb{R}^m_{>0}$,$\mathbf{s} \in \mathbb{R}^m$ we have:*

$$\frac{\rho_\sigma(\mathbf{s})}{\sum_{\mathbf{z}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{Mz})} \leqslant \frac{\rho_\sigma(\mathbf{s})}{\sum_{\mathbf{z}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{s}+\mathbf{Mz})} \leqslant \frac{1}{\sum_{\mathbf{z}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{Mz})}.$$

Proof. Inequality on the left follows directly from [MR07, Lemma 2.9] and the fact that $\mathbf{M\Lambda}$ is an m-dimensional lattice. The inequality on the right is essentially implicit in the proof of [AR04, Lemma 3.2], but for completeness, we give a proof of a slightly generalized statement needed in this work. Let us partition $\mathbf{\Lambda}\backslash\{\mathbf{0}\}$ into two sets $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, such that $\mathbf{x} \in \mathbf{\Lambda}_1$ if and only $-\mathbf{x} \in \mathbf{\Lambda}_2$. Clearly, $|\mathbf{\Lambda}_1| = |\mathbf{\Lambda}_2|$. Then, for $\mathbf{z} \in \mathbf{\Lambda}_1$ we have:

$$\begin{aligned}
\rho_\sigma(\mathbf{s}+\mathbf{Mz}) + \rho_\sigma(\mathbf{s}-\mathbf{Mz}) &= e^{-\sum_{i=1}^m \frac{s_i^2+\langle \mathbf{m}_i,\mathbf{z}\rangle^2}{2\sigma_i^2}} \cdot \left(e^{\sum_{i=1}^m \frac{2\langle \mathbf{m}_i,\mathbf{z}\rangle}{2\sigma_i^2}} + e^{-\sum_{i=1}^m \frac{2\langle \mathbf{m}_i,\mathbf{z}\rangle}{2\sigma_i^2}}\right) \\
&\geqslant 2e^{-\sum_{i=1}^m \frac{s_i^2+\langle \mathbf{m}_i,\mathbf{z}\rangle^2}{2\sigma_i^2}} \\
&\geqslant 2\rho_\sigma(\mathbf{s})\rho_\sigma(\mathbf{Mz}) \\
&\geqslant \rho_\sigma(\mathbf{s}) \cdot (\rho_\sigma(\mathbf{Mz}) + \rho_\sigma(-\mathbf{Mz}))
\end{aligned}$$

where for the first inequality we used the fact that $x + x^{-1} \geqslant 2$ for any $x > 0$. Hence,

$$\begin{aligned}\sum_{\mathbf{z}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{s}+\mathbf{Mz}) &= \rho_\sigma(\mathbf{s}) + \sum_{\mathbf{z}\in\mathbf{\Lambda}_1} (\rho_\sigma(\mathbf{s}+\mathbf{Mz}) + \rho_\sigma(\mathbf{s}-\mathbf{Mz})) \\ &\geqslant \rho_\sigma(\mathbf{s}) + \sum_{\mathbf{z}\in\mathbf{\Lambda}_1} \rho_\sigma(\mathbf{s}) \cdot (\rho_\sigma(\mathbf{Mz}) + \rho_\sigma(-\mathbf{Mz})) \\ &\geqslant \rho_\sigma(\mathbf{s}) \left(1 + \sum_{\mathbf{z}\in\mathbf{\Lambda}_1} (\rho_\sigma(\mathbf{Mz}) + \rho_\sigma(-\mathbf{Mz}))\right) \\ &\geqslant \rho_\sigma(\mathbf{s}) \sum_{\mathbf{z}\in\mathbf{\Lambda}} \rho_\sigma(\mathbf{Mz}).\end{aligned}$$

Thus, the statement holds. □

We are ready to present a theorem that says for which parameters the probability in (16) is upper-bounded by 1/2.

Theorem 3.3. *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{M} \in \mathbb{Z}^{k\times m}$ be arbitrary matrices and denote $\mathbf{m}_i \in \mathbb{Z}^m$ to be the i-th row of $\mathbf{M}$. Furthermore, suppose $\sigma = (\sigma_1, \ldots, \sigma_k)$ satisfies $\sigma_i \geqslant q^{n/m}\sqrt{\frac{ek}{m}}\|\mathbf{m}_i\|_1 + 2$ for $i \in [k]$. Then, for any $\mathbf{s} \in \mathbb{R}^k$, we have:*

$$\frac{\rho_\sigma(\mathbf{s})}{\sum_{\mathbf{z}\in\mathbf{\Lambda}_q^\perp(\mathbf{A})} \rho_\sigma(\mathbf{s}+\mathbf{Mz})} \leqslant \frac{1}{2}.$$

Proof. By Lemma 3.2 we only need to show that $\sum_{\mathbf{z}\in\mathbf{\Lambda}_q^\perp(\mathbf{A})} \rho_\sigma(\mathbf{Mz}) \geqslant 2$. Let us set $\gamma = \left\lceil \frac{1}{2}\sqrt{e}q^{n/m} \right\rceil$ and define the set U as follows:

$$U = \left\{\mathbf{u} \in \mathbf{\Lambda}_q^\perp(\mathbf{A}) \backslash \{\mathbf{0}\} : \|\mathbf{u}\| \leqslant 2\gamma\right\}.$$

First, we lower-bound the cardinality of U. By the pigeonhole principle, there exist at least $\ell + 1 \geqslant (2\gamma)^m/q^n + 1$ vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{\ell+1}$ such that for each $j \in [\ell+1], \|\mathbf{u}_j\|_\infty \leqslant \gamma$ and $\mathbf{Au}_1 = \mathbf{Au}_2 = \ldots = \mathbf{Au}_\ell$. Hence, for all $i \in [\ell]$, we have $\mathbf{u}_i - \mathbf{u}_{l+1} \in U$. Consequently, $|U| \geqslant \ell = (2\gamma)^m/q^n$ and

$$\begin{aligned}\sum_{\mathbf{z}\in\mathbf{\Lambda}_q^\perp(\mathbf{A})} \rho_\sigma(\mathbf{Mz}) &\geqslant 1 + \sum_{\mathbf{z}\in U} \rho_\sigma(\mathbf{Mz}) \\ &\geqslant 1 + \sum_{\mathbf{z}\in U} \exp\left(-\sum_{i=1}^k \frac{\langle \mathbf{m}_i, \mathbf{z}\rangle^2}{2\sigma_i^2}\right) \\ &\geqslant 1 + \sum_{\mathbf{z}\in U} \exp\left(-\sum_{i=1}^k \frac{4\gamma^2\|\mathbf{m}_i\|_1^2}{2\sigma_i^2}\right) \\ &\geqslant 1 + |U| \cdot \exp\left(-\sum_{i=1}^k \frac{2\gamma^2\|\mathbf{m}_i\|_1^2}{\sigma_i^2}\right) \\ &\geqslant 1 + \frac{(2\gamma)^m}{q^n} \exp\left(-\sum_{i=1}^k \frac{2\gamma^2\|\mathbf{m}_i\|_1^2}{\sigma_i^2}\right).\end{aligned}$$

Since, we assumed that $\sigma_i \geqslant 2\gamma\sqrt{k/m}\|\mathbf{m}_i\|_1$ and $\gamma \geqslant \frac{1}{2}\sqrt{e}q^{n/m}$, we obtain:

$$\begin{aligned}\sum_{\mathbf{z}\in\mathbf{\Lambda}_q^\perp(\mathbf{A})} \rho_\sigma(\mathbf{Mz}) &\geqslant 1 + \frac{(2\gamma)^m}{q^n}\exp\left(-\sum_{i=1}^{k}\frac{m}{2k}\right)\\ &\geqslant 1 + \frac{(2\gamma)^m}{q^n}\exp\left(-\frac{m}{2}\right)\\ &\geqslant 1 + \exp\left(\frac{m}{2}\right)\exp\left(-\frac{m}{2}\right)\\ &\geqslant 2\end{aligned}$$

which concludes the proof. □

Finally, to compute the probability in (16), we note that:

$$\Pr[\mathbf{y}^* = \mathbf{y} \wedge \mathbf{s}^* = \mathbf{s}] = D_{\mathbf{\Lambda}_q^\perp(\mathbf{X}),\sigma,\mathbf{c}}(\mathbf{0},\mathbf{0}) = \frac{\rho_\sigma(\mathbf{y},\mathbf{s})}{\rho_\sigma((\mathbf{y},\mathbf{s}) + \mathbf{\Lambda}_q^\perp(\mathbf{X}))}$$

Note for every $\mathbf{u} \in \mathbf{\Lambda}_{\mathcal{R}_q}^\perp(\mathbf{A})$, we have $(\mu\mathbf{u}, -\mathbf{u}) \in \mathbf{\Lambda}_q^\perp(\mathbf{X})$. Therefore,

$$\frac{\rho_\sigma(\mathbf{y},\mathbf{s})}{\rho_\sigma((\mathbf{y},\mathbf{s}) + \mathbf{\Lambda}_q^\perp(\mathbf{X}))} \leqslant \frac{\rho_\sigma(\mathbf{y},\mathbf{s})}{\sum_{\mathbf{u}\in\mathbf{\Lambda}_{\mathcal{R}_q}^\perp(\mathbf{A})}\rho_\sigma\left((\mathbf{y},\mathbf{s}) + \begin{bmatrix}\mu\cdot\mathbf{I}_m\\ -\mathbf{I}_m\end{bmatrix}\mathbf{u}\right)}.$$

Since we set $\sigma_y \geqslant q^{n/m}\sqrt{2e}d + 2$, $\sigma_s \geqslant q^{n/m}\sqrt{2e} + 2$, we can apply Theorem 3.3 for $\sigma = (\sigma_y, \ldots, \sigma_y, \sigma_s, \ldots, \sigma_s) \in \mathbb{R}^{2md}$,

$$\mathbf{A} := \mathsf{Rot}(\mathbf{A}) \in \mathbb{Z}_q^{nd\times md} \text{ and } \mathbf{M} := \mathsf{Rot}\left(\begin{bmatrix}\mu\cdot\mathbf{I}_m\\ -\mathbf{I}_m\end{bmatrix}\right) \in \mathbb{Z}^{2md\times md}.$$

We refer to Appendix B for a more rigorous proof of Theorem 3.1.

4 The blind signature

In this section, we define our blind signature scheme. A blind signature scheme has two parties interacting : a user and a server (or signer), so the user produces a signature under the public key of the server. The security of a blind signature scheme is captured by two properties properly defined in Definitions 2.10 and 2.11 : Blindness and One-More Unforgeability. Blindness informally requires that the server is unable to link a signature to the interaction during which this signature was produced. One-More Unforgeability informally says that after some number ℓ of interactions with the server, the user is not able to produce $\ell + 1$ signatures.

The strategy of our blind signature scheme is as follows : the public key is a collection of N public keys of the one-time signature scheme defined in the previous Section 3. To keep the user/server interaction "blind", the user sends

an encryption of his message, together with a NIZK proof that the ciphertext is well-formed. This encryption scheme is such that the server can homomorphically compute (somewhat efficiently) an encryption of the one-time signature under the i-th public key. This way, the user receives an encryption of a one-time signature of his message, but the response from the server hides its i-th secret key enough so the user can only produce one signature per interaction. Again, to preserve blindness, instead of giving away directly his one-time signature, he gives a NIZK proof of knowledge of a valid one-time signature to one of the public keys.

4.1 Definition of the encryption scheme

In this subsection, we define the first building block of our blind signature scheme : an encryption scheme. This encryption scheme shall be secure against Chosen Plaintext Attacks (we prove in Lemma 4.2 that the distribution of the ciphertext is indistinguishable from uniform) and allow the server to compute a one-time signature of the message while masking his secret key. The proofs of the latter statement is postponed to Section 5. We also define a multi-dimensionnal decryption algorithm Dec for better readability of the blind signature protocol Figure 2.

Notations. Throughout this subsection, we use n, m for dimensions, prime modulus q' and prime p. Please note that the modulus used in the encryption scheme differs from the one we use in the remaining of the blind signature scheme.

Algorithm 1 $\mathsf{KeyGen}()$:

1: $\mathbf{B} \leftarrow \mathcal{R}_{q'}^{n \times m}$
2: $\mathbf{x} \leftarrow \{-\gamma, \ldots, \gamma\}^n$
3: $\mathbf{b}^T = \mathbf{x}^T\mathbf{B} \mod q'$
4: $\mathsf{pk}_{\mathsf{enc}} = (\mathbf{B}, \mathbf{b})$
5: $\mathsf{sk}_{\mathsf{enc}} = \mathbf{x}$
6: **return** $(\mathsf{pk}_{\mathsf{enc}}, \mathsf{sk}_{\mathsf{enc}})$

Algorithm 2 $\mathsf{enc}(\mathsf{pk}_{\mathsf{enc}}, \mu)$:

1: $(\mathbf{r}, \mathbf{e}, e') \leftarrow \{-\gamma, \ldots, \gamma\}^m \times \{-\gamma, \ldots, \gamma\}^n \times \{-\gamma, \ldots, \gamma\}$
2: $\mathbf{t} = p\mathbf{B}\mathbf{r} + p\mathbf{e} \mod q'$
3: $t' = p\mathbf{b}^T\mathbf{r} + pe' + \mu \mod q'$
4: **return** $(\mathbf{t}, t')$

Algorithm 3 $\mathsf{dec}(\mathsf{sk_{enc}}, \mathbf{t}, t')$:

1: $z = t' - \mathbf{x}^T\mathbf{t} \mod q'$
2: **return** $z \mod p$

Theorem 4.1. *The encryption scheme defined through Algorithms 1 to 3 is correct. More precisely, if* $\|\mu\|_\infty \leqslant \lfloor p/2 \rfloor$, $(\mathsf{pk_{enc}}, \mathsf{sk_{enc}}) = \mathsf{KeyGen}()$ *and*

$$(nd\gamma + 1)\gamma \leqslant \frac{q'}{2p} - 1/2, \tag{17}$$

then $\mathsf{dec}(\mathsf{sk_{enc}}, \mathbf{t}, t') = \mu$.

Proof. We compute $z = t' - \mathbf{x}^T\mathbf{t}$. We have :

$$z = p\mathbf{b}^T\mathbf{r} + p\mathbf{e}' + \mu - \mathbf{x}^T(p\mathbf{B}\mathbf{r} + p\mathbf{e}) \tag{18}$$

$$= \mu + p(e' - \mathbf{x}^T\mathbf{e}). \tag{19}$$

Since we assumed $(nd\gamma + 1)\gamma \leqslant \frac{q'}{2p} - 1/2$ and $\|\mu\|_\infty \leqslant p/2$, then $\|\mu + p(e' - \mathbf{x}^T\mathbf{e})\|_\infty \leqslant q'/2$, therefore there is no reduction modulo q' in $\mu + p(e' - \mathbf{x}^T\mathbf{e})$ and hence $z = \mu + p(e' - \mathbf{x}^T\mathbf{e}) \mod p = \mu$. □

Algorithm 4 $\mathsf{Dec}(\mathsf{sk_{enc}}, \mathbf{T}, \mathbf{t}')$:

1: $z^T = \mathbf{t}'^T - \mathbf{x}^T\mathbf{T} \mod q'$
2: **return** $z \mod p$

Lemma 4.2. *Let* μ *be some message,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$, *and* $(\mathbf{t}, t') = \mathsf{enc}(\mathsf{pk}, \mu)$. *Then* $\mathbf{t}, t'$ *is indistinguishable from uniform under* $\mathsf{MLWE}_{m,n-m,S_\gamma}$ *and* $\mathsf{MLWE}_{n+1,m,S_\gamma}$ [7].

Proof. We define a sequence of games.

G_0 : In this game, the adversary $\mathcal{A}$ wins if he distinguishes honest samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$, $(\mathbf{t}, t') = \mathsf{enc}(\mathsf{pk}, \mu)$ from $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$, $(\mathbf{t}, t') \leftarrow \mathcal{R}_q^n \times \mathcal{R}_q$.

G_1 : This game is the same as the previous one, except in the key generation, $\mathbf{b}$ is sampled uniformly random. This game is indistinguishable from the previous one under $\mathsf{MLWE}_{m,n-m,S_\gamma}$.

G_2 : This game is the same as the previous one, except $\mathbf{B}\mathbf{r}$ and $\mathbf{b}^T\mathbf{r}$ are sampled uniformly random. This game is indistinguishable from the previous one under $\mathsf{MLWE}_{n+1,m,S_\gamma}$.

[7] We remind the reader that the encryption scheme's variables and computations are done over $\mathcal{R}_q$, and therefore the MLWE problem is $\mod q$, and S_γ here is those $r \in \mathcal{R}_q$ such that $|r| \leqslant \gamma$.

G_3 : This game is the same as the previous one, except $\mathbf{t}, t'$ are sampled uniformly random. This game is identical to the previous one.

G_4 : This game is the same as the previous one, except the public key is honestly sampled from KeyGen. This game is indistinguishable from the previous one under $\mathsf{MLWE}_{m,n,S_\gamma}$.

The result follows from summing up the advantages. □

4.2 Description of the scheme

We describe in Figure 1 the setup algorithm for the server, the setup algorithm for the user is simply 1) Run KeyGen to generate a key pair for the encryption scheme 2) Run the setup algorithm for the zero-knowledge proofs $\pi_{\mathsf{enc}}, \pi_{\in}$ (notice that the public parameters for π_{enc} and $\pi_{\in}$ must be independent so as to preserve blindness), and finally in Figure 2 we describe our blind signature scheme. The verification algorithm is the verification algorithm of the NIZK $\pi_{\in}$. The scheme contains two verification steps : the verification of the well-formedness of the ciphertext π_{enc} by the server and the verification of valid one-time signature $\mathbf{z}$ such that $\mathbf{Az} = \mu\mathbf{v}_j + \mathbf{w}_j$ by the user. A non-succeeding verification implies abortion of the scheme.

Notations. We let q, q' be a prime moduli (q' is the modulus for the encryption, which will be greater than q, the modulus for the blind signature), p be a prime which shall be smaller than q' but greater than the messages to be encrypted, and N a real number corresponding to the number of blind signatures. We introduce a dimension α which is the height of the public matrix $\mathbf{A}$, and $\sigma_{\mathbf{s}}, \sigma_{\mathbf{y}}, \sigma'$ which are stardard deviations for the Gaussian distributions. One can think of $\sigma_{\mathbf{y}}$ as d times greater than $\sigma_{\mathbf{s}}$ so that $\|\mu\mathbf{s}\| \simeq \|\mathbf{y}\|$. For a matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ and a vector $\mathbf{u} \in \mathcal{R}_q^n$, we write $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A})$ the lattice $\{\mathbf{s} \in \mathcal{R}_q^m \ / \ \mathbf{As} = \mathbf{u}\}$. We omit the subscript $\mathbf{u}$ when $\mathbf{u} = \mathbf{0}$. For the sake of clarity, we write $\sqrt[3]{q} = \lfloor \sqrt[3]{q} \rfloor$. We define a gadget vector $\mathbf{g} = (1 \ \sqrt[3]{q} \ \sqrt[3]{q}^2)$, which we use to define the gadget matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}^T & & & \\ & \mathbf{g}^T & & \\ & & \ddots & \\ & & & \mathbf{g}^T \end{bmatrix}.$$

5 Security proof

In this section, we prove the correctness and security of our blind signature. In Section 5.1, we prove the correctness of the homomorphic computation of the server on the user's ciphertext, from which we infer the correctness of the blind signature scheme. In Section 5.2, we prove the blindness and one-more unforgeability of our blind signature scheme.

Fig. 1. ServerKeyGen() :

$\mathbf{R} \leftarrow (S_1^{\alpha\times\alpha})^{2\times 3}$
$\mathbf{A}' \leftarrow \mathcal{R}_q^{\alpha\times 2\alpha},\ \mathbf{A} = \left[\mathbf{A}' \,|\mathbf{A}'\mathbf{R} - \mathbf{G}\right] \in \mathcal{R}_q^{\alpha\times 5\alpha}$
$\mathsf{seed} \leftarrow \{0,1\}^\lambda,\ (\mathbf{v}_i, \mathbf{w}_i)_{1\leqslant i\leqslant N} = \mathsf{PRNG}(\mathsf{seed})$ (20)
$\mathsf{pk}_{\mathsf{Server}} = (\mathbf{A}, \mathsf{seed}),\ \mathsf{seed}$ which expands through PRNG to $(\mathbf{v}_i, \mathbf{w}_i)_{1\leqslant i\leqslant N}$
$\mathsf{sk}_{\mathsf{Server}} = \mathbf{R}$

5.1 Blind computation on the ciphertext

We first prove in Lemma 5.1 that if both parties follow the protocol honestly, then with overwhelming probability, the user successfully decrypts $\mathbf{z} = \mu\mathbf{s}_i + \mathbf{y}_i$. Next, we prove in Theorem 5.2 that this yields the correctness of the blind signature scheme.

Lemma 5.1. *We use notations from Figure 2. If the user and the server follow the protocol on Figure 2 honestly and if*

$$24\sigma' + d\gamma(n+1) \leqslant \frac{q'}{2p} - \frac{1}{2} \text{ and } 12d\sigma_{\mathbf{s}} + 12\sigma_{\mathbf{y}} \leqslant p/2,$$

then with overwhelming probability $\mathsf{Dec}(\mathsf{sk}_{\mathsf{enc}}, \mathbf{F}, \mathbf{f}')^T = \mathbf{y}_i + \mu\mathbf{s}_i$.

Proof. First, we notice that $\begin{cases} \mathbf{F} = p\mathbf{B}(\mathbf{r}\mathbf{s}_i^T + \mathbf{Y}) + p(\mathbf{e}\mathbf{s}_i^T + \mathbf{Y}') \\ \mathbf{f}^T = p\mathbf{b}(\mathbf{r}\mathbf{s}_i^T + \mathbf{Y}) + p(e'\mathbf{s}_i^T + \mathbf{y}''^T) + \mu\mathbf{s}_i^T + \mathbf{y}_i^T \end{cases}$
Let us write $\bar{\mathbf{Y}} = \mathbf{r}\mathbf{s}_i^T + \mathbf{Y}, \bar{\mathbf{Y}}' = \mathbf{e}\mathbf{s}_i^T + \mathbf{Y}', \bar{\mathbf{y}} = e'\mathbf{s}_i^T + \mathbf{y}''^T$ and $\bar{\mu} = \mu\mathbf{s}_i^T + \mathbf{y}_i^T$. Then, the decryption $\mathsf{Dec}(\mathsf{sk}_{\mathsf{enc}}, \mathbf{F}, \mathbf{f})$ is given by

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}_{\mathsf{enc}}, \mathbf{F}, \mathbf{f}) &= p\mathbf{b}^T\bar{\mathbf{Y}} + p\bar{\mathbf{Y}}' + \bar{\mu} - \mathbf{x}^T(p\mathbf{B}\bar{\mathbf{Y}} + p\bar{\mathbf{Y}}') \mod p \\ &= \bar{\mu} + p(\bar{\mathbf{y}} - \mathbf{x}^T\bar{\mathbf{Y}}') \mod p. \end{aligned}$$

Since we assumed $12d\sigma_{\mathbf{s}} + 12\sigma_{\mathbf{y}} \leqslant p/2$, then with overwhelming probability we have $\|\bar{\mu}\|_\infty \leqslant p/2$. Moreover, we assumed $24\sigma' + d\gamma(n+1) \leqslant \frac{q'}{2p} - \frac{1}{2}$, hence $\|\bar{\mu} + \bar{\mathbf{y}} - \mathbf{x}^T\bar{\mathbf{Y}}'\|_\infty \leqslant \frac{q'}{2p} - \frac{1}{2}$, and therefore $\mathsf{Dec}(\mathsf{sk}_{\mathsf{enc}}, \mathbf{F}, \mathbf{f}) = \bar{\mu}$. □

Theorem 5.2. *The blind signature scheme defined in Figure 2 is correct. More precisely, if both parties follow the protocol honestly, then the produced signature passes verification with overwhelming probability.*

Proof. We only need to prove that the vector $\mathbf{z}$ recovered by the user verifies $\mathbf{A}\mathbf{z} \in \{\mathbf{w}_i + \mu\mathbf{v}_i,\ 1 \leqslant i \leqslant N\}$, so the non-interactive zero-knowledge proof $\pi_\in$ that the user computes as the blind signature passes verification. We chose parameters such that it follows directly from Lemma 5.1 that we have $\mathbf{z} = \mathbf{y}_i + \mu\mathbf{s}_i$, and therefore $\mathbf{A}\mathbf{z} = \mathbf{w}_i + \mu\mathbf{v}_i$ where i is the state of the server when he responded to the user's query. □

Fig. 2. Blind signature scheme

Server		User
State $i, \mathsf{sk}_{\mathsf{Server}} = \mathbf{R}$		message $\mu, \mathsf{sk}_{\mathsf{enc}}$
		$(\mathbf{t}, t') = \mathsf{enc}(\mathsf{pk}_{\mathsf{enc}}, \mu)$
		$\pi_{\mathsf{enc}} = \pi_{\mathsf{enc}}((\mathbf{B}, \mathbf{b}, (\mathbf{t}, t'), (\mathbf{r}, (\mathbf{e}, e'), \mu))$
	$\xleftarrow{\mathbf{t}, t', \pi_{\mathsf{enc}}, \mathsf{pk}_{\mathsf{enc}}}$	
Verify π_{enc}		
$\Lambda_i = \Lambda^{\perp}_{\mathbf{v}_i}(\mathbf{A}),\ \Lambda'_i = \Lambda^{\perp}_{\mathbf{w}_i}(\mathbf{A})$		
$(\mathbf{s}_i, \mathbf{y}_i) \leftarrow D_{\Lambda_i, \sigma_{\mathbf{s}}} \times D_{\Lambda'_i, \sigma_{\mathbf{y}}}$		
$\mathbf{Y}, \mathbf{Y}', \mathbf{y}'' \leftarrow D^{m \times 5\alpha}_{\sigma'} \times D^{n \times 5\alpha}_{\sigma'} \times D^{5\alpha}_{\sigma'}$		(21)
$\mathbf{F} = \mathbf{t}\mathbf{s}_i^T + p\mathbf{B}\mathbf{Y} + p\mathbf{Y}'$		(22)
$\mathbf{f}'^T = \mathbf{y}_i^T + t'\mathbf{s}_i^T + p\mathbf{b}^T\mathbf{Y} + p\mathbf{y}''$		(23)
Update state $i = i + 1$		(24)
	$\xrightarrow{\mathbf{F}, \mathbf{f}'}$	
		$\mathbf{z} = \mathsf{Dec}(\mathsf{sk}_{\mathsf{enc}}, \mathbf{F}, \mathbf{f}')$ (25)
		Verify that $\exists j \leqslant N,\ \mathbf{A}\mathbf{z} = \mu\mathbf{v}_j + \mathbf{w}_j$
		$\mathsf{BlindSig}(\mu) = \pi_{\in}(\mathbf{A}, (\mathbf{w}_j + \mu\mathbf{v}_j)_{1 \leqslant j \leqslant N}, \mathbf{z})$

5.2 Blindness and One-More Unforgeability of the blind signature scheme

The main Theorem of this subsection is Theorem 5.4. We prove blindness directly from a sequence of games, proving that the blindness game is indistinguishable from a game that is independent of the messages. The proof of one-more unforgeability is broken down in 2 parts : first we reduce the one-more unforgeability game to another game OMUF^*. Next, we prove that with rewindable access to an adversary $\mathcal{A}$ with winning probability ϵ, one has probability $O(\epsilon)$ to solve $\mathsf{MSIS}_{\alpha,5\alpha,B}$ for some short bound B.

Lemma 5.3. *Let* OMUF *be the One-More Unforgeability game as described in Definition 2.11. We define a variant of* OMUF *which we call* OMUF^**. The challenger of* OMUF^* *differs from the challenger of* OMUF *only in the key generation. He executes instead the following instructions :*

1. *Generate* $\mathbf{A} \leftarrow \mathcal{R}_q^{\alpha\times 5\alpha}$ *uniformly random*
2. *For* $1 \leqslant i \leqslant N$*, sample* $(\mathbf{s}_i, \mathbf{y}_i) \leftarrow D_{\sigma_\mathbf{s}}^{5\alpha} \times D_{\sigma_\mathbf{y}}^{5\alpha}$
3. *For* $1 \leqslant i \leqslant N$*, compute* $\mathbf{v}_i = \mathbf{A}\mathbf{s}_i$ *and* $\mathbf{w}_i = \mathbf{A}\mathbf{w}_i$
4. *Generate a* seed
5. *Program the* PRNG *so* $\mathsf{PRNG}(\mathsf{seed})$ *expands to* $(\mathbf{v}_i, \mathbf{w}_i)_{1\leqslant i\leqslant N}$
6. *Set the public key of the blind signature scheme to be* $(\mathbf{A}, \mathsf{seed})$.

Then, for any adversary $\mathcal{A}$ *and* ϵ *negligible, if* $\sigma_\mathbf{s}, \sigma_\mathbf{y} \geqslant \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$*, then we have*

$$\epsilon_\mathcal{A}^{\mathsf{OMUF}} \leqslant \epsilon_\mathcal{A}^{\mathsf{OMUF}^*} + \epsilon_\mathcal{A}^{\mathsf{MLWE}_{\alpha,\alpha,S_1}} + 4N\epsilon.$$

Proof. We define a sequence of games :

G_0 : This game is OMUF. The advantage of $\mathcal{A}$ is $\epsilon_\mathcal{A}^{\mathsf{OMUF}}$.

G_1 : This game is the same as the previous one, except $(\mathbf{s}_i, \mathbf{y}_i)_i$ is sampled (from a discrete Gaussian of the same standard deviations $\sigma_\mathbf{s}$ and $\sigma_\mathbf{y}$ as in G_0) and $(\mathbf{v}_i, \mathbf{w}_i)$ is computed. The PRNG is programmed to expand seed onto $(\mathbf{v}_i, \mathbf{w}_i)_i$. Since $\sigma_\mathbf{s}, \sigma_\mathbf{y} \geqslant \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$, it follows from Corollary 2.4 that the distribution of each $(\mathbf{v}_i, \mathbf{w}_i)$ in this game is at statistical distance at most 4ϵ from the distribution of $(\mathbf{v}_i, \mathbf{w}_i)$ in the previous game, for all $1 \leqslant i \leqslant N$. Therefore this game is at distance at most $4N\epsilon$ from G_0.

G_2 : This game is the same as the previous one, except the public matrix $\mathbf{A}$ is sampled uniformly random $\mathbf{A} \leftarrow \mathcal{R}_q^{\alpha\times 5\alpha}$. This game is indistinguishable from G_1 under $\mathsf{MLWE}_{\alpha,\alpha,S_1}$.

The last game G_2 is OMUF^*, hence the adversary $\mathcal{A}$ has advantage $\epsilon_\mathcal{A}^{\mathsf{OMUF}^*}$ against G_2, and the result follows from summing up the advantages. □

The strategy of the one-more unforgeability proof is roughly speaking to rely on the security of our one-time signature from Section 3. More precisely, the reduction $\mathcal{B}$ plays the OMUF^* with $\mathcal{A}$. Similarly as in the unforgeability reduction of the one-time signature, $\mathcal{B}$ knows one preimage $\mu\mathbf{s} + \mathbf{y}$ of $\mu\mathbf{v} + \mathbf{w}$,

and extracts a second one $\mathbf{z}$ from $\mathcal{A}$'s forgery[8]. We cannot argue straight away that $\mathbf{z} - (\mathbf{s}\mu + \mathbf{y})$ is a non-zero solution to MSIS for the public matrix $\mathbf{A}$, since $\mathcal{A}$ may have learnt from the extra information - or hints that $\mathcal{B}$ gave away when sending the ciphertext $(\mathbf{F}, \mathbf{f})$. Indeed, we have

$$\mathbf{F} = \mathbf{t}\mathbf{s}_i^T + p\mathbf{B}\mathbf{Y} + p\mathbf{Y}' = p\mathbf{B}(\mathbf{r}\mathbf{s}_i^T + \mathbf{Y}) + p(\mathbf{e}\mathbf{s}_i^T + \mathbf{Y}')$$
$$\mathbf{f}^T = \mathbf{y}_i + t'\mathbf{s}_i^T + p\mathbf{b}^T\mathbf{Y} + p\mathbf{y}'' = p\mathbf{b}(\mathbf{r}\mathbf{s}_i^T + \mathbf{Y}) + p(e'\mathbf{s}_i^T + \mathbf{y}''^T) + \mathbf{z}.$$

The masks $\mathbf{Y}, \mathbf{Y}', \mathbf{y}''$ hide the secret values $\mathbf{r}\mathbf{s}_i^T, \mathbf{e}\mathbf{s}_i^T, e'\mathbf{s}_i^T$, but we need to take into account the amount of leakage these hints represent. In other word, what is the winning probability of the adversary to the one-time signature when the signer provides hints ? We decided to write Theorem 3.3, which is the foundation of the unforgeability proof of our one-time signature in a general fashion, which encompasses the case with extra hints. The one-more unforgeability proof then boils down to an application of Theorem 3.3.

Theorem 5.4. *The blind signature scheme defined in Figure 2 verifies blindness and one-more unforgeability.*

For blindness, we have the following :

$$\epsilon_{\mathcal{A}}^{blind} \leqslant \epsilon_{\mathcal{A}}^{ZK(\pi_{\mathsf{enc}})} + \epsilon_{\mathcal{A}}^{ZK(\pi_{\in})} + \epsilon_{\mathcal{A}}^{\mathsf{MLWE}_{m,n,S_\gamma}} + \epsilon_{\mathcal{A}}^{\mathsf{MLWE}_{n+m,m,S_\gamma}}.$$

For One-More Unforgeability, if $\sigma_{\mathbf{s}} \geqslant 2+q^{1/5}\sqrt{(m+n+3)}$, $\sigma_{\mathbf{y}} \geqslant 2+q^{1/5}\sqrt{(m+n+3)}d, \sigma' \geqslant 2+q^{1/5}\sqrt{(m+n+3)}\gamma d$ *and* $\mathcal{A}$ *is an adversary with winning probability* ϵ *against* OMUF^**, then there exists an algorithm* $\mathcal{B}$ *that with rewindable black-box access to* $\mathcal{A}$ *can solve* $\mathsf{MSIS}_{\alpha,5\alpha,B}$ *with winning probability at least* $\frac{\epsilon}{2N}$*, where* $B = 24\kappa_{\in}(d\sigma_{\mathbf{s}} + \sigma_{\mathbf{y}}) + 2B_{\in}$*,* $B_{\in}$ *is the bound on the norm verification of the membership proof and* $\delta =$*. This statement combined with Lemma 5.3 gives the One-More Unforgeability of the scheme.*

Proof. Blindness.
We define a sequence of games.

G_0 : This game is the blindness game Definition 2.10. The adversary sends $\mathsf{pk}_{\mathsf{Server}}$ to the challenger $\mathcal{B}$. The challenger runs $\mathsf{UserKeyGen}$ twice. He sends $\mathsf{pk}_0, \mathsf{pk}_1$ to the adversary $\mathcal{A}$. Then, the adversary sends two messages m_0, m_1 of his choice to $\mathcal{B}$, which picks a random bit b. The adversary and the challenger produce $\sigma_0 = \mathsf{BlindSig}(m_0)$ (respectively $\sigma_1 = \mathsf{BlindSig}(m_1)$), and we write $\mathbf{t}_0, t_0', \pi_{\mathsf{enc}}^0, \mathbf{F}_0, \mathbf{f}_0$ (respectively $\mathbf{t}_1, t_1, \pi_{\mathsf{enc}}^1, \mathbf{F}_1, \mathbf{f}_1$) the transcript of their communications. The verification step from the user ensures that the decryption of $\mathbf{F}_0, \mathbf{f}_0$ (respectively $\mathbf{F}_1, \mathbf{f}_1$) is a valid $\mathbf{z}$[9]. The users send (σ_b, σ_{1-b}) to the adversary. The adversary wins if he outputs b.

[8] The forgery is one of the unexpected signatures, which exists since the adversary is expected to produce at most ℓ signatures from ℓ interactions.

[9] Notice that due to this verification step, our definition of blindness is stronger than honest-signer blindness.

G_1 : This game is the same as the previous one, except the challengers runs the simulator of the zero-knowledge proof π_{enc} to produce $\pi^0_{\mathsf{enc}}, \pi^1_{\mathsf{enc}}$. This game is indistinguishable from G_0 under the zero-knowledge property of π_{enc}.

G_2 : This game is the same as the previous one except $\mathbf{t}_0, t'_0, \mathbf{t}_1, t'_1$ are replaced with uniformly random samples. This game is indistinguishable from G_1 under $\mathsf{MLWE}_{m,n,S_\gamma}$ and $\mathsf{MLWE}_{n+m,m,S_\gamma}$ by Lemma 4.2.

G_3 : This game is the same as the previous one, except π^0_ϵ and π^1_ϵ are generated using the simulator from the zero-knowledge proof of π_ϵ. This game is indistinguishable from G_2 under the zero-knowledge property of π_ϵ. This game is independent of b, and therefore, the advantage of $\mathcal{A}$ against G_4 is 0.

The result follows from summing up the advantages.

One-More Unforgeability.

Let $\mathcal{A}$ be an adversary to the OMUF^* game with winning probability ϵ. We describe an efficient algorithm $\mathcal{B}$ that with rewindable black-box access to $\mathcal{A}$ solves $\mathsf{MSIS}_{\alpha,5\alpha,B}$ with $B = 24\kappa_\epsilon(d\sigma_{\mathbf{s}} + \sigma_{\mathbf{y}}) + 2B_\epsilon$.

First, $\mathcal{B}$ receives an $\mathsf{MSIS}_{\alpha,5\alpha,B}$ instance $\mathbf{A} \in \mathcal{R}_q^{\alpha\times 5\alpha}$. Then, $\mathcal{B}$ will execute the following instructions :

1. For $1 \leqslant i \leqslant N$, generate $\mathbf{s}_i, \mathbf{y}_i \leftarrow D^{5\alpha}_{\sigma_{\mathbf{s}}} \times D^{5\alpha}_{\sigma_{\mathbf{y}}}$.
2. For $1 \leqslant i \leqslant N$, set $\mathbf{v}_i = \mathbf{A}\mathbf{s}_i$, $\mathbf{w}_i = \mathbf{A}\mathbf{y}_i$ and set $\mathbf{v}_j = \mathbf{v}$.
3. Sample a random seed.
4. Program the PRNG on input seed such that $\mathsf{PRNG}(\mathsf{seed}) = (\mathbf{v}_i, \mathbf{w}_i)_{1\leqslant i\leqslant N}$.
5. Send the public key $(\mathbf{A}, \mathsf{seed})$ to the adversary $\mathcal{A}$.

Notice that since $\mathbf{A}$ is an $\mathsf{MSIS}_{\alpha,5\alpha,B}$ instance, it is uniformly random and the distribution of the public key that the adversary $\mathcal{A}$ receives is identical to OMUF^*. Next, the adversary sends some number ℓ of queries $(\mathbf{t}_i, t'_i, \pi^i_{\mathsf{enc}})$ to $\mathcal{B}$. The algorithm $\mathcal{B}$ computes honest reponses $(\mathbf{F}_i, \mathbf{f}_i)$ and sends them to $\mathcal{A}$. The adversary has probability at least ϵ to succeed in producing $\ell + 1$ valid signatures, which he sends to $\mathcal{B}$ if he indeed succeeds.

Next, algorithm $\mathcal{B}$ picks a uniformly random index $1 \leqslant j \leqslant \ell + 1$, and runs the extractor $\mathcal{E}$ from the membership proof upon reception of the j-th signature from $\mathcal{A}$. This way, $\mathcal{B}$ extracts an index i, a message μ, a vector $\mathbf{z}$ and a challenge difference $\bar{c}$ such that $\mathbf{A}\mathbf{z} = \bar{c}(\mu\mathbf{v}_i + \mathbf{w}_i)$. We remind that from key generation, $\mathcal{B}$ also knows $\mathbf{z}' = \bar{c}(\mu\mathbf{s}_i + \mathbf{y}_i)$ which verifies the same equation as the extracted $\mathbf{z}$. Three options are possible :

1. The adversary had an interaction with $\mathcal{B}$ on the public key $\mathbf{v}_i, \mathbf{w}_i$ for the message μ, at the end of which the decryption of $\mathcal{B}$'s response $\mathbf{F}_i, \mathbf{f}_i$ is $\mathbf{z}$.

2. The adversary had an interaction with $\mathcal{B}$ on the public key $\mathbf{v}_i, \mathbf{w}_i$ for any message, at the end of which the decryption of $\mathcal{B}$'s response $\mathbf{F}_i, \mathbf{f}_i$ is not $\mathbf{z}$.
3. The adversary never had an interaction with $\mathcal{B}$ on the public $\mathbf{v}_i, \mathbf{w}_i$.

Since $\mathcal{A}$ had ℓ interactions with $\mathcal{B}$ but managed to produce $\ell + 1$ signatures, at least one of these signatures is in option 2) or 3). With probability at least $1/(\ell + 1) \geqslant 1/N$, option 2) or 3) happened, otherwise $\mathcal{B}$ fails and aborts[10].

Option 3) is harder for the adversary than option 2), so we will only deal with the latter. Let us assume that $i, \mu, \mathbf{z}, \bar{c}$ are from option 2). We assume that the adversary $\mathcal{A}$ is able to collect the masks $\mathbf{Y} + \mathbf{r}_i\mathbf{s}_i^T, \mathbf{Y}' + \mathbf{e}\mathbf{s}_i^T, \mathbf{y}'' + e'\mathbf{s}_i^T$. We gather these 3 equations in the form $\boldsymbol{\omega} + \mathbf{P}\mathbf{s}_i = \mathbf{x}$, where $\boldsymbol{\omega}, \mathbf{x} \in \mathcal{R}_q^{5\alpha(m+n+1)}$, and $\mathbf{P}$ is the matrix of the linear function that depends on $\mathbf{r}, \mathbf{e}, e'$ such that $\mathbf{P}\mathbf{s} = (s_1\mathbf{r}\ s_2\mathbf{r}\ \ldots\ s_1\mathbf{e}\ s_2\mathbf{e}\ \ldots\ s_1e'\ s_2e'\ \ldots)$. Both $\mathbf{P}$ and $\mathbf{x}$ are known to the adversary. Since the adversary is able to reconstruct $\mathbf{F}_i, \mathbf{f}_i$ from $\mathbf{X}$, we claim that this assumption is without loss of generality.

The vector $\mathbf{z} - \mathbf{z}'$ (remind that $\mathbf{z}' = \bar{c}(\mu\mathbf{s}_i + \mathbf{y}_i)$ is informally $\mathcal{B}$'s signature of μ times $\bar{c}$) is $\mathcal{B}$'s candidate for $\mathsf{MSIS}_{\alpha,5\alpha,B}$. Indeed,

$$\mathbf{A}(\mathbf{z} - \mathbf{z}') = \bar{c}(\mu\mathbf{v}_i + \mathbf{w}_i) - \bar{c}(\mu\mathbf{A}\mathbf{s}_i + \mathbf{A}\mathbf{y}_i) = 0.$$

Remains to prove i) that the probability that $\mathbf{z} = \mathbf{z}'$ is not negligibly close to 1, and ii) that $\mathbf{z} - \mathbf{z}'$ is shorter than B. First, for i), we introduce the following lattice coset

$$\Lambda = \left\{ (\mathbf{s}, \mathbf{y}, \boldsymbol{\omega}) \in \mathcal{R}_q^{5\alpha} \times \mathcal{R}_q^{5\alpha} \times \mathcal{R}_q^{(n+m+1)5\alpha},\ \begin{cases} \mathbf{A}\mathbf{s} = \mathbf{v}_i \\ \mathbf{A}\mathbf{y} = \mathbf{w}_i \\ \mu_i\mathbf{s} + \mathbf{y}_i = \mathbf{z}_i \\ \mathbf{P}\mathbf{s}^T + \boldsymbol{\omega} = \mathbf{x} \end{cases} \right\}.$$

We claim that all of $\mathcal{A}$'s information on $(\mathbf{s}_i, \mathbf{y}_i)$ is contained in the statement that $(\mathbf{s}_i, \mathbf{y}_i, \boldsymbol{\omega})$ are drawn from χ, which is $D_{\sigma_\mathbf{s}}^{5\alpha} \times D_{\sigma_\mathbf{y}}^{5\alpha} \times D_{\sigma'}^{(m+n+1)5\alpha}$ restricted to Λ. Let $(\mathbf{s}_i', \mathbf{y}_i', \boldsymbol{\omega}')$ be random variables following χ, and let $\boldsymbol{\zeta}(\mathbf{s}_i', \mathbf{y}_i', \boldsymbol{\omega}') = \mu\mathbf{s}_i' + \mathbf{y}_i'$. Notice that for some $\boldsymbol{\zeta}^* \in \mathcal{R}_q^{5\alpha}$, there can be only one tuple $(\mathbf{s}^*, \mathbf{y}^*, \boldsymbol{\omega}^*)$ in the support of χ such that $\boldsymbol{\zeta}^* = \mu\mathbf{s}^* + \mathbf{y}^*$. We have

$$\begin{aligned} \mathbb{P}(\mathbf{z} = \mathbf{z}') &= \mathbb{P}(\mathbf{z} = \boldsymbol{\zeta}(\mathbf{s}_i, \mathbf{y}_i, \boldsymbol{\omega}_i)) \\ &\leqslant \mathbb{P}(\boldsymbol{\zeta}(\mathbf{s}_i', \mathbf{y}_i', \boldsymbol{\omega}') = \boldsymbol{\zeta}(\mathbf{s}_i, \mathbf{y}_i, \boldsymbol{\omega}_i)) \\ &\leqslant \mathbb{P}((\mathbf{s}_i', \mathbf{y}_i', \boldsymbol{\omega}_i') = (\mathbf{s}_i, \mathbf{y}_i, \boldsymbol{\omega}_i)) \\ &\leqslant \chi(\mathbf{s}_i, \mathbf{y}_i, \boldsymbol{\omega}_i). \end{aligned}$$

[10] It seems that $\mathcal{A}$ could send directly the index of the unexpected signature to $\mathcal{B}$. This would save a factor $1/N$ in the winning probability of $\mathcal{B}$ while seemingly keeping the hardness of the forgery the same.

To finish the proof of i), we prove that

$$\max_{(\mathbf{s}'_i,\mathbf{y}'_i,\boldsymbol{\omega}'_i)} \chi(\mathbf{s}'_i,\mathbf{y}'_i,\boldsymbol{\omega}'_i) \leqslant \delta, \tag{26}$$

for some constant δ that is not negligibly close to 1. This fact follows from Theorem 3.3 applied to the rotations of the matrices $\mathbf{A}$, and

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_{5\alpha} \\ -\mu\mathbf{I}_{5\alpha} \\ -\mathbf{P} \end{bmatrix}.$$

The reason is we have

$$\chi(\mathbf{s}^*_i,\mathbf{y}^*_i,\boldsymbol{\omega}^*_i) = \frac{\rho_{\sigma_\mathbf{s}}(\mathbf{s}^*_i)\rho_{\sigma_\mathbf{y}}(\mathbf{y}^*_i)\rho_{\sigma'}(\boldsymbol{\omega}^*_i)}{\sum\limits_{\mathbf{z}=(\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3)\in\Lambda} \rho_{\sigma_\mathbf{s}}(\mathbf{z}_1)\rho_{\sigma_\mathbf{y}}(\mathbf{z}_2)\rho_{\sigma'}(\mathbf{z}_3)}.$$

Now, note that for every $\mathbf{u} \in \Lambda^\perp(\mathbf{A})$, the vector $\mathbf{Mu}$ is such that $(\mathbf{s}^*_i,\mathbf{y}^*_i,\boldsymbol{\omega}^*_i) + \mathbf{Mu} \in \Lambda$. This means that $\mathsf{rot}(M)$ is a valid matrix for Theorem 3.3. If we take $\sigma_\mathbf{s} \geqslant 2 + q^{1/5}\sqrt{(m+n+3)}$, $\sigma_\mathbf{y} \geqslant 2 + q^{1/5}\sqrt{(m+n+3)}d$ and $\sigma' \geqslant 2 + q^{1/5}\sqrt{(m+n+3)}\gamma d$, then Theorem 3.3 ensures $\delta \leqslant 1/2$.

We now prove ii) : $\mathbf{z}$ has the length of the extracted vector from the set membership proof. With $B_\in$ the bound on the norm verification of the membership proof, we have $\|\mathbf{z}\| \leqslant 2B_\in$. On the other hand, $\mathcal{B}$'s private signature $\mathbf{z}' = \bar{c}(\mu\mathbf{s}_i + \mathbf{y}_i)$ is such that $\|\mathbf{z}'\| \leqslant 24\kappa_\in(d\sigma_\mathbf{s} + \sigma_\mathbf{y})$, where $\kappa_\in$ is a bound on the Hamming weight of the challenge difference $\bar{c}$ of the membership proof. Pluging together the inequalities yields ii), which in turn completes the One-More Unforgeability proof. □

Remark on standard deviation bounds. Theorem 3.3 gives lower bounds on the standard deviation of the secrets such that the maximum probability of the secret distribution (which is a multi-dimensionnal Gaussian) is 1/2. As it turns out in our case, there is another lower bound on the standard deviations $\sigma_\mathbf{s}$ and $\sigma_\mathbf{y}$ given by the smoothing parameter for trapdoor sampling, which is greater than the one for one-more unforgeability. Therefore, the actual maximum probability is lower than 1/2, which gives us some more room to decrease the standard deviation σ' of the hints. We leave this remark as a possible optimization of the parameters, that would slightly reduce the communication cost of the blind signature.

6 Parameter Selection

In this section we instantiate our blind signature for at most $N = 2^{18}$ signing queries and aim for 128-bit security (see Fig. 3 and 4). To this end, we measure the hardness of MSIS and MLWE with the root Hermite factor δ and aim for $\delta \approx 1.0043$. For computing hardness of the latter problem, we use the LWE Estimator by Albrecht et al. [APS15]. We refer to [LNS21a, Section 3.3] and [LNS21b, Appendix C] for a detailed explanation on the parameter selection for π_{enc} and $\pi_\in$ respectively.

Parameter	Definition	Instantiation
N	maximum number of signing queries	2^{18}
d	dimension of the ring $\mathcal{R}$	128
q	modulus for the blind signature	$\approx 2^{64}$
q'	modulus for the encryption	$\approx 2^{128}$
α	height of the matrix $\mathbf{A} = [\mathbf{A}'\|\mathbf{A}'\mathbf{R} - \mathbf{G}]$	21
σ_y	standard deviation for sampling $\mathbf{y}$	$\approx 2^{30}$
σ_s	standard deviation for sampling $\mathbf{s}$	$\approx 2^{30}$
n	height of the encryption public key matrix $\mathbf{B}$	80
m	width of the encryption public key matrix $\mathbf{B}$	40
γ	maximum coefficient of the $\mathbf{x}, \mathbf{r}$ and errors $\mathbf{e}, e'$	4
p	additional prime number, less than q', used for encryption	$\approx 2^{43}$
σ'	standard deviation used to sample maskings $\mathbf{Y}, \mathbf{Y}'$ and $\mathbf{y}''$	$\approx 2^{26}$

Fig. 3. Definition and concrete numbers for parameters used in the blind signature construction.

Dimensions and moduli. Firstly, we choose the ring dimension $d = 128$ and moduli $(q, q') = (\approx 2^{64}, \approx 2^{128})$ [11]. Next, we want to make sure that $\mathbf{A} = [\mathbf{A}'|\mathbf{A}'\mathbf{R} - \mathbf{G}]$ is indistinguishable from a random matrix over $\mathcal{R}_q$. Hence, we choose $\alpha = 21$ such that $\mathsf{MLWE}_{\alpha,\alpha,S_1}$ is hard. Then, in order to apply Micciancio-Peikert trapdoor sampling [MP12], we need the standard deviations σ_s, σ_y to be at least $2(s_1(\mathbf{R}) + 1)\sqrt{\lceil q^{2/3} \rceil + 1}$ where s_1 is the operator norm. Similarly as in [dPLS18, Section 2.6], we found experimentally that for a structured matrix $\mathbf{R} \in S_1^{2\alpha \times 3\alpha}$, $s_1(\mathbf{R}) \leqslant 6\sqrt{\alpha d}$ with a high probability. Note that the other lower bound for σ_y, σ_s in Theorem 5.4 is smaller than the one necessary for trapdoor sampling. Hence, in this scenario we will set $\sigma := \sigma_s = \sigma_y = 13\sqrt{\alpha d \left(\lceil q^{2/3} \rceil + 1\right)}$.

Encryption scheme. We now focus on parameters for the encryption scheme. In order to ensure the property that both the public key and the ciphertext are indistinguishable from random, we need $\mathsf{MLWE}_{m,n-m,S_\gamma}$ and $\mathsf{MLWE}_{n+1,m,S_\gamma}$ to be hard. We set $n = 2m$ and thus these two problems are almost equally hard. Since $q' \approx 2^{128}$, we pick $(n, m) = (80, 40)$ and $\gamma = 4$. Then we set $p = 12d\sigma_s + 12\sigma_y$ and $\sigma' = 2 + q^{1/5}\sqrt{(m + n + 3)}\gamma d$. For such a large $q' \approx 2^{128}$, correctness conditions from Lemma 5.1 follow easily.

Verifiable encryption. We turn to computing the proof sizes for π_{enc} and $\pi_{\in}$. Let us focus on the former one first. Let $\tilde{n} := m + (n + 1) + 1$ be the number polynomials in the vector $(\mathbf{r}, \mathbf{e}, e', \mu)$ and $\tilde{\alpha} = 2\gamma + 1$ [12]. Then, in order to prove

[11] More specifically, we choose $q \approx 2^{64}$ for which $X^d + 1$ splits into quadratic terms modulo q. This makes sure the one-out-of-many proof $\pi_{\in}$ from [LNS21b] does not need any repetitions.

[12] Intuitively, $\tilde{\alpha}$ represents how many garbage polynomials we need to prove that coefficients a polynomial are exactly between $-\gamma$ and γ. For example, if one wants to prove ternary coefficients, we need three garbage polynomials.

π_{enc} (see Section 2.6), we apply the framework from [LNS21a]. As discussed in [LNS21a, Section 3.3], the proof with soundness error $1/q' \approx 2^{-128}$, i.e. no repetitions, has size upper-bounded by:

$$(\tilde{n} + \tilde{\kappa} + \tilde{\alpha} + 1)d \log q' + (\tilde{\lambda} + \tilde{n} + \tilde{\kappa} + \tilde{\alpha})d \log(12\mathfrak{s}) \text{ bits}^{13}.$$

The standard deviation $\mathfrak{s}$ is set as $\mathfrak{s} = d\sqrt{(\tilde{\lambda} + \tilde{n} + \tilde{\kappa} + \tilde{\alpha})d}$. Then, $\tilde{\kappa}$ and $\tilde{\lambda}$ are chosen such that $\mathsf{MSIS}_{\tilde{\kappa},\tilde{\lambda}+\tilde{n}+\tilde{\kappa}+\tilde{\alpha},8d\beta}$ and $\mathsf{MLWE}_{\tilde{n}+\tilde{\kappa}+\tilde{\alpha},\tilde{\lambda},\chi^d}$ are hard [14], where $\beta = \mathfrak{s}\sqrt{2(\tilde{\lambda} + \tilde{n} + \tilde{\kappa} + \tilde{\alpha})d}$ and χ is the distribution on $\{-1, 0, 1\}$ where ± 1 both have probability 5/16 and 0 has probability 6/16. To further reduce the proof size, we apply the Dilithium compression described in [LNS21a, Appendix B].

Communication complexity. In order to compute total communication size, we calculate the total size of public key and ciphertexts sent by both the user $\mathcal{U}$ and the signer $\mathcal{S}$. Note that $\mathcal{U}$ sends $m + n + 1$ elements in $\mathcal{R}_q$. On the other hand, $\mathcal{S}$ outputs back $5\alpha(n + 1)$ polynomials. Hence, the total communication size, excluding π_{enc}, is

$$(m + (5\alpha + 1)(n + 1)) \log q' \text{ bits}.$$

Signature size. Finally, to estimate the signature size, we need to look at the one-out-of-many proof $\pi_{\in}$. Let us set $m' = 2$, i.e. $(\log q)^{m'+1} = 2^{18} = N$. As described in [LNS21b, Appendix C], the proof size of $\pi_{\in}$ can be bounded by:

$$(\kappa' + \alpha + 2m' + 2)d \log q + 5\alpha d \log(12\mathfrak{s}') + (\kappa' + \lambda' + \alpha + 2m' + 2)d \log(12\mathfrak{s}'')$$

bits. We set $\mathfrak{s}' = d(d+1)\sigma\sqrt{10\alpha d}$ and $\mathfrak{s}'' = d\sqrt{(\kappa' + \lambda' + \alpha + 2m' + 2)d}$. Then, κ' and λ' are chosen such that $\mathsf{MSIS}_{\kappa',\kappa'+\lambda'+\alpha+2m'+2,8d\beta''}$ and $\mathsf{MLWE}_{\kappa'+\alpha+2m'+2,\lambda',\chi^d}$ are hard where $\beta'' = \mathfrak{s}''\sqrt{2(\kappa' + \lambda' + \alpha + 2m' + 2)d}$. Eventually, in order to ensure one-more-unforgeability, we check that $\mathsf{MSIS}_{\alpha,5\alpha,2\mathfrak{s}'\sqrt{2(\kappa'+\lambda'+\alpha+2m'+2)d}}$ is a hard problem. As before, we apply the Dilithium compression technique when computing the signature/proof size.

Reducing the public key size. We observe that the public key contains the matrix $\mathbf{A}'\mathbf{R}$ which cannot be generated from the seed. It consists of $3\alpha^2$ polynomials in $\mathcal{R}_q$ and for parameters selected above, the total public key size is above 1MB as presented in Fig. 4. In order to reduce the public key size, we apply the technique by Lyubashevsky et al. [LNPS21] where one can decrease the value

[13] For simplicity, we neglect the size of a challenge polynomial since it has a negligible impact on the total proof size.

[14] Actually, the zero-knowledge property of the protocol in [LNS21a] reduces to the so-called Extemded-MLWE problem. However, as argued in [LNS21a], this problem should still be almost as hard as the plain MLWE.

Public key	Secret key	Signature	Communication
1.3MB	75KB	150KB	16MB

Fig. 4. Public key, user secret key, signature sizes and communication complexity of our blind signature scheme.

of α at the cost of increasing the ring dimension d[15]. Then, one observes that the equations over $\mathcal{R}_q$ which we are interested in, can be equivalently written over the ring $\mathbb{Z}_q[X]/(X^{128}+1)$ and then proven using e.g. [ALS20, LNS21a]. However, as a drawback of having a large ring dimension, we would obtain slightly larger signatures and communication complexity.

Acknowledgements

We would like to thank anonymous reviewers for the useful feedback. This work was supported by the EU H2020 ERC Project 101002845 PLAZA.

References

ABB20. Nabil Alkeilani Alkadri, Rachid El Bansarkhani, and Johannes Buchmann. BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In *Financial Cryptography*, volume 12059 of *Lecture Notes in Computer Science*, pages 484–502. Springer, 2020.

ALS20. Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In *CRYPTO (2)*, volume 12171 of *Lecture Notes in Computer Science*, pages 470–499. Springer, 2020.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. `https://eprint.iacr.org/2015/046`.

AR04. Dorit Aharonov and Oded Regev. Lattice problems in NP cap conp. In *FOCS*, pages 362–371. IEEE Computer Society, 2004.

ASY21. Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Towards practical and round-optimal lattice-based threshold and blind signatures. *IACR Cryptol. ePrint Arch.*, page 381, 2021.

Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, Dec 1993.

BLL+15. Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the rényi divergence rather than the statistical distance. In *ASIACRYPT (1)*, volume 9452 of *Lecture Notes in Computer Science*, pages 3–24. Springer, 2015.

BLL+21. Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In *EUROCRYPT (1)*, volume 12696 of *Lecture Notes in Computer Science*, pages 33–53. Springer, 2021.

[15] For instance, when $\alpha = 1$ and $d = 4096$, the public key has size $\approx$ 300KB.

BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 176–202. Springer, 2019.

Cha82. David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203. Plenum Press, New York, 1982.

DKL+18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

dPLS18. Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *ACM Conference on Computer and Communications Security*, pages 574–591. ACM, 2018.

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT (2)*, pages 259–288, 2020.

ESLL19. Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 115–146. Springer, 2019.

EZS+19. Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Matrict: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *CCS*, pages 567–584. ACM, 2019.

GK15. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT*, pages 253–280, 2015.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

HKLN20. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In *CRYPTO (2)*, volume 12171 of *Lecture Notes in Computer Science*, pages 500–529. Springer, 2020.

LM18. Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. *J. Cryptol.*, 31(3):774–797, 2018. Preliminare version appeared in TCC 2008.

LNPS21. Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *ASIACRYPT (4)*, volume 13093 of *Lecture Notes in Computer Science*, pages 218–248. Springer, 2021.

LNS20. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In *CCS*, pages 1051–1070. ACM, 2020.

LNS21a. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *Public Key Cryptography (1)*, volume 12710 of *Lecture Notes in Computer Science*, pages 215–241. Springer, 2021.

LNS21b. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. SMILE: set membership from ideal lattices with applications to ring signatures and confidential transactions. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 611–640. Springer, 2021.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.
Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.
MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.
MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
PFH+17. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, , and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2017. https://csrc.nist.gov/projects/post-quantum-cryptography/ round-1-submissions.
PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
Rüc10. Markus Rückert. Lattice-based blind signatures. In *ASIACRYPT*, pages 413–430, 2010.
YAZ+19. Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 147–175. Springer, 2019.

A Proof of Corollary 2.4

Take arbitrary $\mathbf{x}^* \in \mathbb{R}^n$ and $\mathbf{t}^* \in \mathbf{\Lambda}\backslash\mathbf{\Lambda}'$. If $\mathbf{t}^* \neq \mathbf{x}^* \bmod \mathbf{\Lambda}'$ then clearly $D_1(\mathbf{x}^*, \mathbf{t}^*) = D_2(\mathbf{x}^*, \mathbf{t}^*) = 0$. Hence, assume that $\mathbf{t}^* = \mathbf{x}^* \bmod \mathbf{\Lambda}'$.

First, we observe that $D_1(\mathbf{x}^*, \mathbf{t}^*) = D_{\mathbf{\Lambda},\sigma}(\mathbf{x}^*)$. On the other hand, we have

$$\begin{aligned} D_2(\mathbf{x}^*, \mathbf{t}^*) &= \Pr\left[\mathbf{x} = \mathbf{x}^* | \mathbf{t} = \mathbf{t}^*\right] \cdot \Pr\left[\mathbf{t} = \mathbf{t}^*\right] \\ &= \Pr\left[\mathbf{x} = \mathbf{x}^* | \mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right] \cdot \frac{1}{|\mathbf{\Lambda}\backslash\mathbf{\Lambda}'|} \\ &= \frac{\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}' | \mathbf{x} = \mathbf{x}^*\right] \cdot \Pr\left[\mathbf{x} = \mathbf{x}^*\right]}{\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right]} \cdot \frac{1}{|\mathbf{\Lambda}\backslash\mathbf{\Lambda}'|} \\ &= \frac{D_{\mathbf{\Lambda},\sigma}(\mathbf{x}^*)}{\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right]} \cdot \frac{1}{|\mathbf{\Lambda}\backslash\mathbf{\Lambda}'|}. \end{aligned}$$

We want to estimate $\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right]$. By Lemma 2.3 we have

$$\rho_\sigma(\mathbf{\Lambda}) = \sum_{\mathbf{t} \in \mathbf{\Lambda}\backslash\mathbf{\Lambda}'} \rho_\sigma(\mathbf{t} + \mathbf{\Lambda}') = \sum_{\mathbf{t} \in \mathbf{\Lambda}\backslash\mathbf{\Lambda}'} \rho_{\sigma,-\mathbf{t}}(\mathbf{\Lambda}') \leqslant \sum_{\mathbf{t} \in \mathbf{\Lambda}\backslash\mathbf{\Lambda}'} \rho_\sigma(\mathbf{\Lambda}') = |\mathbf{\Lambda}\backslash\mathbf{\Lambda}'| \cdot \rho_\sigma(\mathbf{\Lambda}').$$

Similarly, we obtain $\rho_\sigma(\mathbf{\Lambda}) \geqslant \frac{1-\varepsilon}{1+\varepsilon} \cdot |\mathbf{\Lambda}\backslash\mathbf{\Lambda}'| \cdot \rho_\sigma(\mathbf{\Lambda}')$. Then, again by Lemma 2.3:

$$\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right] = \frac{\rho_{\sigma,-\mathbf{t}^*}(\mathbf{\Lambda}')}{\rho_\sigma(\mathbf{\Lambda})} \geqslant \frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{\rho_\sigma(\mathbf{\Lambda}')}{\rho_\sigma(\mathbf{\Lambda})} \geqslant \frac{1-\varepsilon}{1+\varepsilon} \cdot \frac{1}{|\mathbf{\Lambda}\backslash\mathbf{\Lambda}'|}$$

and

$$\Pr\left[\mathbf{t}^* = \mathbf{x} \bmod \mathbf{\Lambda}'\right] \leqslant \frac{\rho_\sigma(\mathbf{\Lambda}')}{\rho_\sigma(\mathbf{\Lambda})} \leqslant \frac{1+\varepsilon}{1-\varepsilon} \cdot \frac{1}{|\mathbf{\Lambda}\backslash\mathbf{\Lambda}'|}.$$

This implies that

$$D_2(\mathbf{x}^*, \mathbf{t}^*) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot D_{\mathbf{\Lambda},s}(\mathbf{x}^*)$$

and thus

$$|D_1(\mathbf{x}^*, \mathbf{t}^*) - D_2(\mathbf{x}^*, \mathbf{t}^*)| \leqslant \frac{2\varepsilon}{1-\varepsilon} D_{\mathbf{\Lambda},\sigma}(\mathbf{x}^*) \leqslant 4\varepsilon D_{\mathbf{\Lambda},\sigma}(\mathbf{x}^*).$$

Finally, we conclude that

$$\Delta(D_1, D_2) \leqslant \sum_{\mathbf{t}^* \in \mathbf{\Lambda} \setminus \mathbf{\Lambda}'} \sum_{\mathbf{x}^* \in \mathbf{t}^* + \mathbf{\Lambda}'} 4\varepsilon D_{\mathbf{\Lambda},\sigma}(\mathbf{x}^*) = 4\varepsilon \sum_{\mathbf{t}^* \in \mathbf{\Lambda} \setminus \mathbf{\Lambda}'} D_{\mathbf{\Lambda},\sigma}(\mathbf{t}^* + \mathbf{\Lambda}') = 4\varepsilon.$$

B Proof of Theorem 3.1

In order to prove correctness, we observe that for $\mathbf{y} \leftarrow D_{\sigma_y}^{md}, \mathbf{s} \leftarrow D_{\sigma_s}^{md}$ and $\mu \in \mathcal{R}_q$ being a polynomial with binary coefficients we have

$$\|\mathbf{y} + \mu\mathbf{s}\| \leqslant \|\mathbf{y}\| + \|\mu\mathbf{s}\| \leqslant \|\mathbf{y}\| + d\|\mathbf{s}\| \leqslant (\sigma_y + d\sigma_s)\sqrt{2md}$$

with probability at least

$$1 - 2 \cdot \left(\frac{2}{e}\right)^{\frac{md}{2}} > 1 - 2 \cdot 2^{-\frac{md}{6}} > 1 - 2^{\lambda}.$$

by applying Lemma 2.5 for $k = \sqrt{2}$. Finally, $\mathbf{z} := \mathbf{y} + m\mathbf{s}$ satisfies:

$$\mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{y} + \mu\mathbf{A}\mathbf{s} = \mathbf{w} + \mu\mathbf{v}.$$

For the unforgeability argument, we follow the proof strategy from [LM18]. Suppose there is an adversary $\mathcal{A}$ which succeeds in breaking the strong unforgeability game of the one-time signature scheme with probability γ. We use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ which solves MSIS. Namely, $\mathcal{B}$ does the following:

1. Given a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, it samples $\mathbf{y} \leftarrow D_{\sigma_y}^{md}, \mathbf{s} \leftarrow D_{\sigma_s}^{md}$ and sets $\mathbf{w} = \mathbf{A}\mathbf{y}, \mathbf{v} = \mathbf{A}\mathbf{s}$. Finally, $\mathcal{B}$ outputs $(\mathbf{A}, \mathbf{v}, \mathbf{w})$ to $\mathcal{A}$.
2. Once $\mathcal{A}$ queries on input $\mu \in \{0,1\}^d$, $\mathcal{B}$ returns $\mathbf{z} = \mathbf{y} + \mu\mathbf{s}$.
3. At the end, $\mathcal{A}$ outputs a forgery $(\tilde{\mu}, \tilde{\mathbf{z}})$. If $\mathbf{A}\tilde{\mathbf{z}} \neq \mathbf{w} + \tilde{\mu}\mathbf{v}$ or $\|\tilde{\mathbf{z}}\| > (\sigma_y + d\sigma_s)\sqrt{2md}$, then $\mathcal{B}$ manually sets $\tilde{\mu} = \mu$ and $\tilde{\mathbf{z}} = \mathbf{z}$. Then, $\mathcal{B}$ outputs a MSIS solution $\mathbf{y} + \tilde{\mu}\mathbf{s} - \tilde{\mathbf{z}}$.

Let us define the following events:

$$\mathsf{success} = (\mathbf{y} + \tilde{\mu}\mathbf{s} - \tilde{\mathbf{z}} \neq \mathbf{0}) \wedge (\|\mathbf{y} + \tilde{\mu}\mathbf{s} - \tilde{\mathbf{z}}\| \leqslant 2(\sigma_y + d\sigma_s)\sqrt{2md}$$

and

$$\mathsf{forgery} = ((\tilde{\mu}, \tilde{\mathbf{z}}) \neq (\mu, \mathbf{z}))\,.$$

By assumption, we have that $\Pr[\mathsf{forgery}] = \gamma$. Clearly, $\mathbf{y} + \tilde{\mu}\mathbf{s} - \tilde{\mathbf{z}}$ is a MSIS solution to $\mathbf{A}$. Hence, we are interested in computing the probability of $\mathsf{success}$. To do so, we consider a similar algorithm $\mathcal{B}^*$ which does the following:

1. Given a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n\times m}$, it samples $\mathbf{y} \leftarrow D_{\sigma_y}^{md}, \mathbf{s} \leftarrow D_{\sigma_s}^{md}$ and sets $\mathbf{w} = \mathbf{A}\mathbf{y}, \mathbf{v} = \mathbf{A}\mathbf{s}$. Finally, $\mathcal{B}$ outputs $(\mathbf{A}, \mathbf{v}, \mathbf{w})$ to $\mathcal{A}$.
2. Once $\mathcal{A}$ queries on input $\mu \in \{0,1\}^d$, $\mathcal{B}$ picks a bit b such that $\Pr[b = 0] = 1/3$ and $\Pr[b = 1] = 2/3$. If $b = 0$, it sets $\mathbf{y}^* = \mathbf{y}$ and $\mathbf{s}^* = \mathbf{s}$. Otherwise, B^* samples $(\mathbf{y}^*, \mathbf{s}^*) \in D_\sigma^{2md}$ conditioned on $\mathbf{A}\mathbf{y}^* = \mathbf{A}\mathbf{y}, \mathbf{A}\mathbf{s}^* = \mathbf{A}\mathbf{s}$ and $\mathbf{y}^* + \mu\mathbf{s}^* = \mathbf{y} + \mu\mathbf{s}$. In other words, $(\mathbf{y}^*, \mathbf{s}^*) \leftarrow D_{\mathbf{\Lambda}_q^\perp(\mathbf{X}),\sigma,\mathbf{c}}$ where
$$\mathbf{X} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \\ 1 \cdot \mathbf{I}_m & \mu \cdot \mathbf{I}_m \end{pmatrix} \in \mathcal{R}_q^{3m\times 2m} \text{ and } \mathbf{c} = -\begin{pmatrix} \mathbf{y} \\ \mathbf{s} \end{pmatrix} \in \mathcal{R}_q^{2m}.$$
Then, it returns $\mathbf{z} = \mathbf{y} + \mu\mathbf{s}$ to $\mathcal{A}$.
3. At the end, $\mathcal{A}$ outputs a forgery $(\tilde{\mu}, \tilde{\mathbf{z}})$. If $\mathbf{A}\tilde{\mathbf{z}} \neq \mathbf{w} + \tilde{\mu}\mathbf{v}$ or $\|\tilde{\mathbf{z}}\| > (\sigma_y + d\sigma_s)\sqrt{2md}$, then $\mathcal{B}$ manually sets $\tilde{\mu} = \mu$ and $\tilde{\mathbf{z}} = \mathbf{z}$. Then, $\mathcal{B}$ outputs a MSIS solution $\mathbf{y}^* + \tilde{\mu}\mathbf{s}^* - \tilde{\mathbf{z}}$.

Note that the algorithm $\mathcal{B}^*$ is not necessarily efficient. However, the key distribution of $(\mathbf{y}^*, \mathbf{s}^*)$ is identical to $(\mathbf{y}, \mathbf{s})$ in $\mathcal{B}$ given $\mathbf{A}, \mathbf{v}, \mathbf{w}, \mu$ and $\mathbf{z}$. Thus, the output distribution of $\mathcal{B}^*$ is identical to the one by $\mathcal{B}$. In particular, if we define the following event:
$$\mathsf{success}^* = (\mathbf{y}^* + \tilde{\mu}\mathbf{s}^* - \tilde{\mathbf{z}} \neq \mathbf{0}) \wedge (\|\mathbf{y}^* + \tilde{\mu}\mathbf{s}^* - \tilde{\mathbf{z}}\| \leqslant (\sigma_y + d\sigma_s)\sqrt{2md}$$
then we have $\Pr[\mathsf{success}] = \Pr[\mathsf{success}^*]$ and similarly, the event $\mathsf{forgery}^* = ((\tilde{\mu}, \tilde{\mathbf{z}}) \neq (\mu, \mathbf{z}))$ defined for $\mathcal{B}^*$ satisfies $\Pr[\mathsf{forgery}^*] = \gamma$.

For the remainder of the proof, we assume that $\mathbf{y}, \mathbf{y}^*$ and $\mathbf{s}, \mathbf{s}^*$ have L_2 norm bounded by $\sigma_y\sqrt{2md}$ and $\sigma_s\sqrt{2md}$ respectively. Indeed, the probability that one of these vectors has the norm larger than these bounds is negligible by Lemma 2.5. Then, we can already drop the event $\|\mathbf{y}^* + \tilde{\mu}\mathbf{s}^* - \tilde{\mathbf{z}}\| \leqslant 2(\sigma_y + d\sigma_s)\sqrt{2md}$ from the definition of $\mathsf{success}^*$ since
$$\|\mathbf{y}^* + \tilde{\mu}\mathbf{s}^* - \tilde{\mathbf{z}}\| \leqslant \|\mathbf{y}^*\| + d\|\mathbf{s}^*\| + \|\tilde{\mathbf{z}}\| \leqslant 2(\sigma_y + d\sigma_s)\sqrt{2md}.$$

Now, we break the probability $\Pr[\mathsf{success}^*]$ into three parts:
$$\begin{aligned} \Pr[\mathsf{success}^*] = &\Pr[\mathsf{success}^* \wedge \mu = \tilde{\mu}] \\ &+ \Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s}] \\ &+ \Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}]. \end{aligned}$$

In the following, we will use the observation that the bit b is independent of $\mathbf{z}, \tilde{\mathbf{z}}, \mu, \tilde{\mu}$ and $\mathbf{y}, \mathbf{s}$.

We focus on upper-bounding the first term. Note that
$$\begin{aligned} \Pr[\mathsf{success}^* \wedge \mu = \tilde{\mu}] &\geqslant \Pr[\mathsf{success}^* \wedge \mu = \tilde{\mu} \wedge b = 0] \\ &\geqslant \Pr[\tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s} \wedge \mu = \tilde{\mu}] \cdot \frac{1}{3} \\ &\geqslant \Pr[(\mu, \mathbf{z}) \neq (\tilde{\mu}, \tilde{\mathbf{z}}) \wedge \mu = \tilde{\mu}] \cdot \frac{1}{3} \\ &\geqslant \Pr[\mathsf{forgery}^* \wedge \mu = \tilde{\mu}] \cdot \frac{1}{3}. \end{aligned} \tag{27}$$

For the second term, we observe that:

$$\begin{aligned}\Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s}] \\ &\geqslant \Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s} \wedge b = 0] \\ &\geqslant \Pr[\mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{1}{3} \\ &\geqslant \Pr[(\mu, \mathbf{z}) \neq (\tilde{\mu}, \tilde{\mathbf{z}}) \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{1}{3} \\ &\geqslant \Pr[\mathsf{forgery}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} \neq \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{1}{3}.\end{aligned} \tag{28}$$

Finally, we focus on the last term. For readability, let us define the event E as:

$$E = (\mu \neq \tilde{\mu}) \wedge (\tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}) \wedge (b = 1).$$

Then, we have

$$\begin{aligned}\Pr[E] &\geqslant \Pr[\mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{2}{3} \\ &\geqslant \Pr[\mathsf{forgery}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{2}{3}\end{aligned}$$

and also

$$\Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}] \geqslant \Pr[\mathbf{y} + \tilde{\mu}\mathbf{s} \neq \mathbf{y}^* + \tilde{\mu}\mathbf{s}^* | E] \cdot \Pr[E].$$

Hence, we need to upper-bound $\Pr[\mathbf{y} + \tilde{\mu}\mathbf{s} = \mathbf{y}^* + \tilde{\mu}\mathbf{s}^* | E]$. First, we claim that

$$\Pr[\mathbf{y} + \tilde{\mu}\mathbf{s} = \mathbf{y}^* + \tilde{\mu}\mathbf{s}^* | E] = Pr[\mathbf{y} = \mathbf{y}^* \wedge \mathbf{s} = \mathbf{s}^* | E].$$

Clearly, $\mathbf{y} = \mathbf{y}^* \wedge \mathbf{s} = \mathbf{s}^*$ implies $\mathbf{y} + \tilde{\mu}\mathbf{s} = \mathbf{y}^* + \tilde{\mu}\mathbf{s}^*$. Now, suppose that given E and $\mathbf{y} + \tilde{\mu}\mathbf{s} = \mathbf{y}^* + \tilde{\mu}\mathbf{s}^*$ hold. By definition of $\mathbf{y}^*$ and $\mathbf{s}^*$ we have $\mathbf{y}^* + \mu\mathbf{s}^* = \mathbf{z} = \mathbf{y} + \mu\mathbf{s}$. Thus, $(\mu - \tilde{\mu})(\mathbf{s} - \mathbf{s}^*) = \mathbf{0}$. Now, by assumption we have $\|\mathbf{s} - \mathbf{s}^*\| \leqslant 2\sigma_s\sqrt{2md}$. Therefore

$$\|(\mu - \tilde{\mu})(\mathbf{s} - \mathbf{s}^*)\|_\infty \leqslant 2d\sigma_s\sqrt{2md} < q/2.$$

We conclude that $(\mu - \tilde{\mu})(\mathbf{s} - \mathbf{s}^*) = \mathbf{0}$ over $\mathcal{R}$. Since we assumed that $\mu \neq \tilde{\mu}$, we must have $\mathbf{s} = \mathbf{s}^*$. Then, we also get $\mathbf{y} = \mathbf{y}^*$.

Now, we want to upper-bound $\Pr[\mathbf{y}^* = \mathbf{y} \wedge \mathbf{s}^* = \mathbf{s} | E]$, i.e.

$$\Pr[\mathbf{y}^* = \mathbf{y} \wedge \mathbf{s}^* = \mathbf{s} | E] = D_{\mathbf{\Lambda}_q^\perp(\mathbf{X}), \sigma, \mathbf{c}}(\mathbf{0}, \mathbf{0}) = \frac{\rho_\sigma(\mathbf{y}, \mathbf{s})}{\rho_\sigma((\mathbf{y}, \mathbf{s}) + \mathbf{\Lambda}_q^\perp(\mathbf{X}))}$$

Note for every $\mathbf{u} \in \mathbf{\Lambda}_{\mathcal{R}_q}^\perp(\mathbf{A})$, we have $(\mu\mathbf{u}, -\mathbf{u}) \in \mathbf{\Lambda}_q^\perp(\mathbf{X})$. Therefore,

$$\frac{\rho_\sigma(\mathbf{y}, \mathbf{s})}{\rho_\sigma((\mathbf{y}, \mathbf{s}) + \mathbf{\Lambda}_q^\perp(\mathbf{X}))} \leqslant \frac{\rho_\sigma(\mathbf{y}, \mathbf{s})}{\sum_{\mathbf{u} \in \mathbf{\Lambda}_{\mathcal{R}_q}^\perp(\mathbf{A})} \rho_\sigma\left((\mathbf{y}, \mathbf{s}) + \begin{bmatrix} \mu \cdot \mathbf{I}_m \\ -\mathbf{I}_m \end{bmatrix} \mathbf{u}\right)}.$$

Since we set $\sigma_y \geqslant q^{n/m}\sqrt{2e}d + 2$, $\sigma_s \geqslant q^{n/m}\sqrt{2e} + 2$, we can apply Theorem 3.3 for $\sigma = (\sigma_y, \ldots, \sigma_y, \sigma_s, \ldots, \sigma_s)$,

$$\mathbf{A} := \mathsf{Rot}(\mathbf{A}) \in \mathbb{Z}_q^{nd \times md} \text{ and } \mathbf{M} := \mathsf{Rot}\left(\begin{bmatrix} \mu \cdot \mathbf{I}_m \\ -\mathbf{I}_m \end{bmatrix}\right) \in \mathbb{Z}^{2md \times md16}.$$

Thus, we conclude that

$$\Pr[\mathbf{y} = \mathbf{y}^* \wedge \mathbf{s} = \mathbf{s}^* | E] \leqslant \frac{1}{2}$$

and therefore

$$\Pr[\mathsf{success}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}] \geqslant \Pr[\mathsf{forgery}^* \wedge \mu \neq \tilde{\mu} \wedge \tilde{\mathbf{z}} = \mathbf{y} + \tilde{\mu}\mathbf{s}] \cdot \frac{1}{3}. \quad (29)$$

By combining Equations 27,28 and 29, we obtain:

$$\begin{aligned} \Pr[\mathsf{success}^*] &\geqslant \frac{1}{3} \cdot \Pr[\mathsf{forgery}^*] - \mathsf{negl}(\lambda) \\ &\geqslant \frac{1}{3}\gamma - \mathsf{negl}(\lambda). \end{aligned}$$

[16] We need to work with rotation matrices since Theorem 3.3 only considers matrices over integers.