A New Framework For More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling

Rafael del Pino[1] and Shuichi Katsumata[2]

[1] PQShield SAS, France
rafael.del.pino@pqshield.com
[2] AIST, Japan and PQShield Ltd., U.K.
shuichi.katsumata@aist.go.jp

Abstract. Blind signatures, proposed by Chaum (CRYPTO'82), are interactive protocols between a signer and a user, where a user can obtain a signature without revealing the message to be signed. Recently, Hauck et al. (EUROCRYPT'20) observed that all efficient lattice-based blind signatures following the blueprint of the original blind signature by Rükert (ASIACRYPT'10) have a flawed security proof. This puts us in a situation where all known lattice-based blind signatures have at least two of the following drawbacks: heuristic security; 1 MB or more signature size; only supporting bounded polynomially many signatures, or being based on non-standard assumptions.

In this work, we construct the first *round-optimal* (i.e., two-round) lattice-based blind signature with a signature size roughly 100 KB that supports unbounded polynomially many signatures and is provably secure under standard assumptions. Even if we allow non-standard assumptions and more rounds, ours provide the shortest signature size while simultaneously supporting unbounded polynomially many signatures. The main idea of our work is revisiting the generic blind signature construction by Fischlin (CRYPTO'06) and optimizing the *commit-then-open* proof using techniques tailored to lattices. Our blind signature is also the first construction to have a formal security proof in the *quantum* random oracle model. Finally, our blind signature extends naturally to *partially* blind signatures, where the user and signer can include an agreed-upon public string in the message.

1 Introduction

1.1 Background

Blind signatures, originally proposed by Chaum [22], are interactive protocols between a signer and a user, where a user can obtain a signature without revealing the message to be signed to the signer. Blind signatures satisfy two security notions: *one-more unforgeability* and *blindness*. One-more unforgeability states that if a malicious user engages only in at most ℓ (possibly concurrent) signing

sessions with the signer, then it cannot output more than ℓ signatures. Blindness states that a malicious signer can neither learn the message during the signing session nor link a particular message-signature pair to a particular signing session. The typical applications of blind signatures include e-cash [22, 24, 42], anonymous credentials [18, 20], e-voting [23, 31], and so on, and more recently, it has found exciting applications in the context of adding privacy features to blockchains [49] and privacy-preserving authentication tokens [1].

In this paper, we focus on one class of blind signatures that has recently attracted a lot of attention: *lattice-based* blind signatures; currently the only known class of blind signatures believed to withstand quantum attacks for other related works). The first lattice-based blind signature was proposed by Rükert [46], who followed a design paradigm similar to the classical Schnorr or Okamoto-Schnorr blind signatures [47, 44]. The blind signature consists of three rounds and supports poly-logarithmically many signatures (in the security parameter λ) before having to regenerate the verification key. This general approach has been extended and optimized in subsequent works [43, 35, 8, 9, 7], where BLAZE+ by Alkadri et al. [9] currently stands as the most efficient proposal. However, recently, Hauck et al. [33] showed that all constructions following the blueprint of Rükert's blind signature contain the same bug in their security proof[3], consequently leaving them only heuristically secure at best. Building on Rükert's blind signature and optimizations employed by BLAZE+, Hauck et al. managed to construct the first provably secure lattice-based blind signature. Unfortunately, the security proof required very large parameter sets, and their proposal resulted in a signature size of roughly 7.9 MB with a communication cost of 34 MB and supported only 7 signatures per verification key. Thus, the work of Hauck et al. [33] reopened the question of building efficient *and* provably secure lattice-based blind signatures.

Very recently, two works aimed at solving this. One by Agrawal et al. [5]. Instead of following the three-move structure seen in Schnorr's blind signature [47], Agrawal et al. builds on Fischlin [30] and Garg et al. [32] that provide a generic construction of a two-move (i.e., *round-optimal*) blind signatures. Concretely, they propose two constructions. One produces a short signature in the range of a few KB with a communication cost of around 50 MB but comes with several caveats: the scheme can support only bounded polynomially many signatures; blindness only holds against *very honest* signers (i.e. the public key must be generated honestly and the signer cannot deviate from the protocol), and the scheme is only heuristically secure as it needs to homomorphically evaluate a standard signature scheme that internally uses a hash function modeled as a random oracle. The second can support unbounded polynomially many signatures and blindness holds against *honest* signers (i.e. the public key must be generated honestly but the signer can deviate from the protocol) but it requires a new non-standard hardness assumption called the *one-more*-inhomogeneous

[3] Alkadri et al. [7] claims to have fixed the bug of BLAZE+ (and thus by Rükert) but we have found several errors in their security proof. This has been confirmed by the authors through personal communication.

SIS assumption. Moreover, the signature size becomes as large as 1 MB[4][5], while the communication cost is lowered to a few KB. The other work is by Lyubashevsky et al. [38]. They propose a round-optimal blind signature based on a new approach using one-time signatures and OR-proofs. Unlike [5], the security of their blind signature is based on the standard hardness of the MSIS and MLWE assumptions. However, the scheme only supports bounded polynomially many signatures with a signature size of roughly 150 KB. The communication cost is around 16 MB and the signer running time scales linearly in the maximum number of signatures that can be signed.

In summary, all known lattice-based blind signatures have at least two of the following drawbacks: heuristic security; 1 MB or more signature size; only supporting bounded polynomially many signatures, or based on non-standard assumptions. This leaves open the following natural question:

> *Can we construct an efficient and provably secure lattice-based blind signature supporting unbounded polynomially many signatures based on standard assumptions?*

As an independent interest, we also note that all provably secure lattice-based blind signatures mentioned above are only proven secure against classical adversaries in the classical random oracle model (ROM). Indeed, most strategies used to prove security completely break down when handling quantum adversaries in the quantum ROM (QROM). Although we do not imagine all previous constructions can be broken using quantum adversaries, considering that one of the main appeals of lattice-based cryptography is their resilience against quantum adversaries, we believe any formal post-quantum security guarantee is highly desirable.

1.2 Our Contribution

In this work, we answer the above question in the affirmative. We construct the first round-optimal lattice-based blind signature with a signature size roughly 100 KB that supports unbounded many signatures and is provably secure under standard assumptions. Even if we allow non-standard assumptions and more rounds, ours provide the shortest signature size while also supporting unbounded many signatures. The communication cost currently sits at 850 KB, but as we explain later, we believe by using the right non-interactive zero-knowledge (NIZK)

[4] Agrawal et al. provide an informal estimate of 30 KB to 100 KB and states to use the NIZK by [28, 40]. However, considering that their security proof relies on an *exact* proof for a relation $\mathbf{Cs} = u$ for a large matrix $\mathbf{C}$ (since the authors argue that $\mathbf{C}$ is indistinguishable from uniform with the leftover hash lemma) and a witness s with entries as large as $\Omega(\sqrt{q})$, even an optimistic estimate gives a lower bound of 1 MB with current lattice-based NIZKs.

[5] After submission of this paper, Agrawal et al. updated their paper to use the NIZK by Lyubashevsky et al. [39] appearing at CRYPTO 2022. See **??** work for more detail.

proofs, we could cut this down to roughly 100 KB while maintaining the same signature size. The security of our blind signature is established both in the classical ROM and QROM. It is secure against *malicious* signers, where blindness holds even when the signer can register malicious keys and deviate from the protocol. Moreover, our scheme can be easily transformed into a *partially* blind signature [2]. This allows the user and signer to include a common agreed-upon message into the signature and has proven to be useful in applications such as e-cash [22, 24, 42] and e-voting [23, 31].

We obtain our blind signature by a new generic construction tailored to lattices. The starting point of our work is the generic round-optimal blind signature construction by Fischlin [30]. The signature in Fischlin's blind signature consists of a complex NIZK proof that informally proves possession of two things: a signature from a standard signature scheme and an opening to a commitment. At the heart of our generic construction is a technique inspired by del Pino et al. [26] that allows us to transform such complex statement into a simple lattice statement consisting only of proving possession of a short vector. Consequently, we can rely on well-known efficient lattice-based NIZKs such as those by Lyubashevsky [36, 37] to generate the signature.

One tool required by our generic construction is a *multi-proof straight-line extractable* NIZK [15],[6] which is used by the user to prove the well-formedness of its first message sent to the signer. Informally, such an NIZK guarantees the existence of an extractor that, on input a simulation trapdoor and any adaptively chosen proofs, outputs the corresponding witnesses. This is in sharp contrast to standard NIZKs in the (Q)ROM where witness extraction is performed via rewinding [44, 13]. If we were to rely on rewinding-based extractions, our security proof would incur an exponential security loss in the number of signing sessions, and result in a scheme that can only support poly-logarithmically many signatures. Similar issues crop up in the context of IND-CCA secure public key encryptions [48, 14] and group signatures [15]. In this work, to construct such strong NIZKs for relatively complex lattice-based statements, we rely on the recent technique of *extractable linear homomorphic commitments* proposed by Katsumata [34].

Finally, we highlight that due to the modularity of our generic construction, any future improvements in lattice-based NIZKs may lead to more efficient blind signatures. For instance, if we were able to combine the technique of Katsumata with the recent efficient lattice-based NIZKs [10, 28], then we could potentially reduce the communication cost from 850 KB to roughly 100 KB. We leave further optimized instantiations of our generic construction as an interesting future work.

1.3 Technical Overview

We give an overview of our techniques in two parts. In Part 1, we explain the high level idea of our generic construction and in Part 2, we explain how to instantiate the building blocks.

[6] This notion is also called *online* extractable in the literature.

Part 1. We first explain our generic construction tailored to lattices.

Blind Signature by Fischlin. Our starting point is the generic construction of blind signatures by Fischlin [30]. The blind signature is round optimal and supports polynomially many signatures. His generic construction relies on general NIZKs for a complex statement and the proof overhead (i.e. signature size) becomes prohibitively large when instantiated using known lattice-based NIZKs. Our goal is to replace this complex statement with a lattice-friendly statement.

We first recall Fischlin's construction. In his construction, the signer publishes a verification key of a standard signature scheme as the verification key vk of the blind signature and keeps the corresponding signing key sk secret. If a user wants the signer to blindly sign on message M, it submits a commitment $\mathsf{com} \leftarrow \mathsf{Com}(\mathsf{M}; \mathsf{rand})$ to the signer and obtains a signature $\sigma \xleftarrow{\$} \mathsf{Sig}(\mathsf{sk}, \mathsf{com})$. The user then constructs a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, \mathsf{com} \| \mathsf{rand} \| \sigma; \mathsf{rand}')$ using a PKE scheme and constructs an NIZK proof π that proves

$$\begin{aligned} \mathsf{com} = \mathsf{Com}(\mathsf{M}; \mathsf{rand}) \ &\wedge \ \mathsf{Verify}(\mathsf{vk}, \sigma, \mathsf{com}) = \top \\ &\wedge \ \mathsf{ct} = \mathsf{Enc}(\mathsf{ek}, \mathsf{com} \| \mathsf{rand} \| \sigma; \mathsf{rand}'), \end{aligned} \tag{1}$$

where the statement is $(\mathsf{vk}, \mathsf{ek}, \mathsf{ct}, \mathsf{M})$ and the witness is $(\mathsf{com}, \mathsf{rand}, \sigma, \mathsf{rand}')$. Finally, the user outputs $\Sigma = (\pi, \mathsf{ct})$ as the blind signature. Here, we assume ek is pseudorandom and is generated as an output of the random oracle. This ensures that nobody, including a malicious signer, knows the corresponding decryption key dk of the PKE scheme in the real-world. dk is only used during the security proof of one-more unforgeability, where the reduction uses dk to decrypt $\mathsf{com} \| \mathsf{rand} \| \sigma$ from ct.

Although it is theoretically possible to instantiate Fischlin's generic construction from lattices, the main bottleneck is constructing an efficient lattice-based NIZK for Eq. (1). Agrawal et al. [5] attempts to heuristically[7] instantiate Fischlin's generic construction based on Dilithium [27], one of the most efficient lattice-based signatures, but they estimated the signature to require at least 100KB with prover complexity approaching 1 hour.

Lattice-Friendly Enc-then-Prove by del Pino et al. The main complexity of Eq. (1) comes from the need to show possession of a valid signature on a hidden message (i.e. com). Roughly, this is because we do not have a lattice-based signature whose verification algorithm is compatible with known efficient lattice-based NIZKs. Now, although not exactly what we require, we observe that a technique used by del Pino et al. [26] for constructing efficient group signatures comes close to what we need.

A group signature allows a user to anonymously sign on behalf of a group, while a special entity called a group manager can deanonymize the signer should the need arise. A typical recipe for constructing a group signature is the *enc-then-prove* paradigm [19]. Each group user is assigned an identity $I \in [N]$, where

[7] Their NIZK requires evaluating a hash function used by Dilithium which is modeled as a random oracle. Considering that a random oracle does not have a function description in the ROM, this approach fails to provide any form of provable security.

$N = \mathsf{poly}(\lambda)$ is the size of the group, and the group manager provides a signature $\sigma \xleftarrow{\$} \mathsf{Sign}(\mathsf{sk}, I)$; this serves as a certificate for user I belonging to the group. To sign on behalf of the group, user I constructs a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{ek}, I; \mathsf{rand}')$ using a PKE scheme and constructs an NIZK proof π that proves

$$\mathsf{Verify}(\mathsf{vk}, \sigma, I) = \top \ \wedge \ \mathsf{ct} = \mathsf{Enc}(\mathsf{ek}, I; \mathsf{rand}'), \tag{2}$$

where the statement X_{GS} is $(\mathsf{vk}, \mathsf{ek}, \mathsf{ct})$ and the witness W_{GS} is $(\sigma, I, \mathsf{rand}')$. Note that NIZKs based on the Fiat-Shamir paradigm allows to bind any message M to a proof π so π indeed serves as a signature for M. Although Eq. (2) seems simpler than Eq. (1), it serves our purpose since it still includes the most complex component, which is proving a valid signature on a hidden message (i.e. I).

We briefly go over the group signature by del Pino et al. [26]. They use Boyen's lattice-based signature [17, 4] as the underlying signature scheme. In Boyen's signature, the verification key consists of a random element $u \in R_q$ and vectors $(\mathbf{a}_1, \mathbf{a}_2) \in R_q^k \times R_q^k$, where R_q is the polynomial ring $\mathbb{Z}_q[X]/(X^d + 1)$. The signing key sk is a short basis $\mathbf{T}_{\mathbf{a}_1} \in R^{k \times k}$ such that $\mathbf{a}_1 \mathbf{T}_{\mathbf{a}_1} = \mathbf{0} \mod q$. To give out a credential for user $I \in [N]$, the group manager views I as a message and samples, using sk, a short vector $\mathbf{e} \in R^{2k}$ satisfying

$$[\mathbf{a}_1 | \mathbf{a}_2 + I \cdot \mathbf{g}] \mathbf{e}^\top = u, \tag{3}$$

where $\mathbf{g}$ is the so-called gadget matrix [41]. It outputs $\mathbf{e}$ as the certificate for user I belonging to the group. If I can be made public, then a user can simply use a standard lattice-based NIZK for proving $\mathsf{MSIS}/\mathsf{MLWE}$ relations to prove possession of the certificate $\mathbf{e}$. That is, relations of the form $\overline{\mathbf{a}}\, \overline{\mathbf{e}}^\top = \overline{u}$, where $(\overline{\mathbf{a}}, \overline{u})$ is the statement and $\overline{\mathbf{e}}$ is the witness. On the other hand, if I needs to be kept private, which is the case for group signatures, then Eq. (3) becomes a quadratic relation over the witness and we no longer know how to prove it efficiently using lattice-based NIZKs.

The technical novelty of del Pino et al. was to linearize Eq. (3) by using the commitment scheme by Baum et al. [12], a.k.a., the BDLOP commitment. The BDLOP commitment is of the form $\mathsf{com} = \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \mathbf{R} + \begin{bmatrix} \mathbf{0} \\ I \cdot \mathbf{g} \end{bmatrix}$, where $\mathbf{b}_0, \mathbf{b}_1 \in R_q^k$ is the commitment key, $\mathbf{R} \in R^{k \times k}$ is the commitment randomness, and $I \cdot \mathbf{g}$ is the message. This commitment satisfies binding and hiding based on the MSIS and MLWE assumptions. Using the lower half of the commitment $\mathbf{t}_1$, we can rewrite the left hand side of Eq. (3) as

$$\begin{aligned} [\mathbf{a}_1 | \mathbf{a}_2 + I \cdot \mathbf{g}] \mathbf{e}^\top &= [\mathbf{a}_1 | \mathbf{a}_2 + \mathbf{b}_1 \mathbf{R} + I \cdot \mathbf{g}]\, \mathbf{e}^\top - \mathbf{b}_1 \mathbf{R} \mathbf{e}_2^\top \\ &= [\mathbf{a}_1 | \mathbf{a}_2 + \mathbf{t}_1 | \mathbf{b}_1] \begin{bmatrix} \mathbf{e}^\top \\ -\mathbf{R}\mathbf{e}_2^\top \end{bmatrix}, \end{aligned} \tag{4}$$

where $\mathbf{e} = [\mathbf{e}_1 | \mathbf{e}_2] \in R^{2k}$. Notice that $[\mathbf{a}_1 | \mathbf{a}_2 + \mathbf{t}_1 | \mathbf{b}_1]$ consists only of public elements included in the statement X_{GS}. Specifically, Eq. (3) can now be expressed as an MSIS relation where the statement is $[\mathbf{a}_1 | \mathbf{a}_2 + \mathbf{t}_1 | \mathbf{b}_1]$ and the witness vector

is $[\mathbf{e}| - \mathbf{e}_2\mathbf{R}^\top] \in R^{3k}$. Thus, the user transforms Eq. (3) into Eq. (4), constructs an efficient NIZK proof π for Eq. (4), and finally outputs the group signature $\Sigma = (\pi, \mathsf{com})$.[8]

Reversing the Order for Blind Signatures. The technique of del Pino et al. [26] can be seen as transforming a Boyen signature on message M into a signature on a commitment com of M. This is a good fit for the group signature functionality; a group authority signs the message $\mathsf{M} = I$ in the clear and the user can later prove possession of the signature while hiding its identity I by planting a commitment com.

Our idea is to turn this technique around and use it for blind signatures. Blind signature has an opposite functionality; the signer signs the message blindly through a commitment and the user later unblinds the commitment to prove possession of a signature. Concretely, a user first constructs a BDLOP commitment com for a message $I \in [N]$ and sends it to the signer.[9] The signer then pulls out $\mathbf{t}_1 \in R_q^k$ included in com and signs $\mathbf{t}_1$ with the Boyen signature. Specifically, the signer samples a short vector $\mathbf{e} \in R^{2k}$ satisfying

$$[\mathbf{a}_1|\mathbf{a}_2 + \mathbf{t}_1]\mathbf{e}^\top = u.$$

The user then reverses the transformation in Eq. (4) to obtain

$$[\mathbf{a}_1|\mathbf{a}_2 + \mathbf{t}_1]\,\mathbf{e}^\top = [\mathbf{a}_1|\mathbf{a}_2 + \mathbf{b}_1\mathbf{R} + I \cdot \mathbf{g}]\,\mathbf{e}^\top = [\mathbf{a}_1|\mathbf{a}_2 + I \cdot \mathbf{g}|\mathbf{b}_1] \begin{bmatrix} \mathbf{e}^\top \\ \mathbf{R}\mathbf{e}_2^\top \end{bmatrix}, \quad (5)$$

where notice the right hand side has the desired form of a public vector being multiplied by a short secret vector. Therefore, the signature output by the user can be a standard NIZK proof π for the MSIS relation, where the statement is $[\mathbf{a}_1|\mathbf{a}_2 + I \cdot \mathbf{g}|\mathbf{b}_1]$ and the witness vector is $[\mathbf{e}|\mathbf{e}_2\mathbf{R}^\top] \in R^{3k}$.

While the above construction satisfies correctness and blindness, it is not clear how to prove one-more unforgeability. To explain why, let us first see how del Pino et al. showed the unforgeability of their group signature. The reduction simulates the group manager by sampling $\mathbf{a}_1 \xleftarrow{\$} R_q^k$ and programming $\mathbf{a}_2$ as $\mathbf{a}_2 = \mathbf{a}_1\mathbf{R}^* - I^* \cdot \mathbf{g}$ for a random short matrix $\mathbf{R}^*$, where $I^* \in [N]$ is a guess for the user on which the adversary forges on. When the adversary queries the certificate for some user $I \neq I^*$, the reduction can use standard techniques [3, 21] to sample a short vector for $[\mathbf{a}_1|\mathbf{a}_2 + I \cdot \mathbf{g}] = [\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^* + (I - I^*) \cdot \mathbf{g}]$ using the simulation trapdoor $\mathbf{R}^*$ and the fact that $(I - I^*)$ is invertible over R_q. Once the adversary outputs a forgery, which consists of a proof π and commitment $\mathbf{t}_1$ satisfying Eq. (4), the reduction (roughly) extracts a witness $(I', \mathbf{R}', \mathbf{e}')$ via rewinding the adversary. By soundness of the NIZK, the witness satisfies $\mathbf{t}_1 = \mathbf{b}_1\mathbf{R}' + I' \cdot \mathbf{g}$ (i.e.

[8] To be precise, the user also needs to prove additional relations, e.g., com is a commitment to some $I \in [N]$. Since these details are not relevant to the core idea, we omit them.

[9] A keen reader may notice that the message space (i.e. group size) $[N]$ has to be polynomial large for the security proof of [26] to work. We later show how to support an exponentially large message space as required for blind signatures.

a valid BDLOP commitment) and

$$[\mathbf{a}_1|\mathbf{a}_2+\mathbf{t}_1|\mathbf{b}_1]\,\mathbf{e}'^{\top}=\left[\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^*-I^*\cdot\mathbf{g}+\mathbf{b}_1\mathbf{R}'+I'\cdot\mathbf{g}|\mathbf{b}_1\right]\mathbf{e}'^{\top}=[\mathbf{a}_1|\mathbf{b}_1]\begin{bmatrix}\mathbf{e}_1'^{\top}+\mathbf{R}^*\mathbf{e}_2'^{\top}\\ \mathbf{R}'\mathbf{e}_2'^{\top}+\mathbf{e}_3'^{\top}\end{bmatrix},$$

where $\mathbf{e}'=[\mathbf{e}_1'|\mathbf{e}_2'|\mathbf{e}_3']\in R^{3k}$ and we assume the guess made by the reduction is correct, i.e. $I^*=I'$, which happens with non-negligible probability when $N=\mathsf{poly}(\lambda)$. Thus, the reduction can break the MSIS problem with respect to the public vector $[\mathbf{a}_1|\mathbf{b}_1]$ if the adversary breaks unforgeability.

Unfortunately, this proof strategy fails in the blind signature setting. In the group signature setting, the reduction only had to sample from the vector $[\mathbf{a}_1|\mathbf{a}_2+I\cdot\mathbf{g}]=[\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^*+(I-I^*)\cdot\mathbf{g}]$, where $I\in[N]$ was the only component controlled by the adversary. However, in the blind signature setting, the reduction must be able to sample from the vector $[\mathbf{a}_1|\mathbf{a}_2+\mathbf{t}_1]=[\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^*-I^*\cdot\mathbf{g}+\mathbf{t}_1]$ for an arbitrary $\mathbf{t}_1$. This change no longer allows the reduction to rely on prior trapdoor sampling techniques [3, 21] and it is not obvious anymore how to simulate the real-world signer without the full trapdoor $\mathbf{T}_{\mathbf{a}_1}$.

Adding Proof of Wellformedness. To fix the above idea, we modify the user to also include an NIZK proof π_{com} of the fact that com is well-formed, which in particular implies that $\mathbf{t}_1=\mathbf{b}_1\mathbf{R}'+I'\cdot\mathbf{g}$ for some short $\mathbf{R}'$ and $I'\in[N]$. However, this cannot be just any standard NIZK. When the reduction is given the proof π_{com} and com from the adversary, it must extract $(\mathbf{R}',I')$ from it without interrupting the simulation. This is in contrast to rewinding-type extractions [44, 13], where the reduction performs extraction only after the adversary finished playing the security game. For example, recall above to see how the reduction extracted an MSIS solution from the adversary's forgery in the unforgeability proof of the group signature. To this end, as we have already pointed to in Sec. 1.2, we rely on a stronger type of *multi-proof straight-line extractable* NIZK [15]. Such NIZK allows the reduction to directly extract $(\mathbf{R}',I')$ from the adversary without altering its behavior.

In summary, the high level description of our blind signature is as follows. The user first constructs a BDLOP commitment com for the message M and adds a multi-proof straight-line extractable NIZK proof π_{com} of its well-formedness. The signer receives $(\pi_{\mathsf{com}},\mathsf{com})$ from the user and then samples a short vector $\mathbf{e}$ such that $[\mathbf{a}_1|\mathbf{a}_2+\mathbf{t}_1|\mathbf{b}_1]\mathbf{e}^{\top}=u$, where notice that we modify the public vector to also include $\mathbf{b}_1$. Given $\mathbf{e}$ from the signer, the user transforms the signature verification equation into an MSIS relation following almost the same computation as in Eq. (5), and outputs a standard NIZK proof π for the MSIS relation as its signature.

In the security proof, the reduction uses the multi-proof straight-line extractable NIZK to extract $(\mathbf{R}',I')$ such that $\mathbf{t}_1=\mathbf{b}_1\mathbf{R}'+I'\cdot\mathbf{g}$ without rewinding the adversary. Then, it can rewrite $[\mathbf{a}_1|\mathbf{a}_2+\mathbf{t}_1|\mathbf{b}_1]$ as $[\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^*+\mathbf{b}_1\mathbf{R}'+(I'-I^*)\cdot\mathbf{g}|\mathbf{b}_1]$. Since $(\mathbf{R}^*,\mathbf{R}')$ serves as a simulation trapdoor for $[\mathbf{a}_1|\mathbf{b}_1]$, the reduction is able to sample a short vector using prior techniques [3, 21] when $I'\neq I^*$. If the adversary outputs a forgery on message I^*, the reduction can obtain an MSIS

solution following an argument similar to that of del Pino et al. This completes the high-level description of our blind signature.

Omitted Details. As we briefly mentioned in Footnote 9, the above proof only works when the message space $[N]$ is polynomially large, which was the only case required in the context of group signatures. Here, if N was larger than polynomial, the probability that the reduction guesses the message I^* output by the adversary becomes negligible. To support an exponential message space, we hash the message I onto a carefully chosen exponential-sized set and sign the hashed message instead. If the hash function is modeled as a random oracle, then the reduction will be able to guess the *hash* of the message used in the forgery with non-negligible probability. Although this simple idea no longer works in the QROM since the adversary can query the entire input space in superposition, we rely on the programming technique of Zhandry [50] to prove security.

Another subtle yet important detail we glossed over is the fact that typical lattice-based NIZKs do not allow for *exact* extraction/soundness. Namely, the reduction may only be able to extract a witness $(\mathbf{R}', I')$ such that $\widehat{c}\cdot\mathbf{t}_1 = \mathbf{b}_1\mathbf{R}' + I'\cdot\mathbf{g}$ from the malicious user, where $\widehat{c}$ is some small invertible element in R_q. In this case, $[\mathbf{a}_1|\mathbf{a}_2 + \mathbf{t}_1|\mathbf{b}_1]$ can only be rewritten as $[\mathbf{a}_1|\mathbf{a}_1\mathbf{R}^* + \mathbf{b}_1(\mathbf{R}'/\widehat{c}) + (I'/\widehat{c} - I^*)\cdot\mathbf{g}|\mathbf{b}_1]$, where $\widehat{c}^{-1}$ is in general not small. Then, since the trapdoor $(\mathbf{R}^*, \mathbf{R}'/\widehat{c})$ is not necessarily small, it no longer fits the description required by prior trapdoor sampling techniques [3, 21]. We show that prior sampling techniques can be naturally extended to work for this setting.

Part 2. Our generic construction relies on two NIZKs for different statements. One is a multi-proof straight-line extractable NIZK used by the user to prove the well-formedness of the first message, i.e. BDLOP commitment. The other is a standard NIZK for the MSIS relation that only needs to be single-proof extractable via rewinding, which is used by the user to construct the final blind signature. We only explain the former as it is the more technically challenging NIZK to construct.

To construct a multi-proof straight-line extractable NIZK, we rely on the recent Katsumata transform [34]. At a high level, it provides a generic method to upgrade many of the known lattice-based NIZKs proven to be secure in the classical ROM to NIZKs secure in the QROM. More precisely, this transform can be seen as a technique to upgrade a single-proof *rewinding*-extractable lattice-based NIZK in the classical ROM into a single-proof *straight-line* extractable NIZK in the QROM. We show that using a more fine-grained analysis, we can further upgrade this transform to provide the desired *multi-proof* straight-line extractable NIZK in the QROM. Thus, the question boils down to constructing a lattice-based NIZK in the classical ROM that is compatible with the Katsumata transform.

Recall the statement we need to prove was roughly $\mathbf{t}_1 = \mathbf{b}_1\mathbf{R} + \mathsf{M}\cdot\mathbf{g}$ with witness $(\mathbf{R}, \mathsf{M})$, where $(\mathbf{R}, \mathsf{M})$ are short/small elements over R_q. A standard way to prove such relation is to first decompose the statement into $(t_{1,i} = \mathbf{b}_1\mathbf{r}_i^\top + \mathsf{M}\cdot g_i)_{i\in[k]}$, where $t_{1,i}, g_i$ and $\mathbf{r}_i$ are the i-th elements and column of $\mathbf{t}_1, \mathbf{g}$, and $\mathbf{R}$,

respectively. By rewriting each $\mathbf{b}_1\mathbf{r}_i^\top + \mathsf{M} \cdot g_i$ into an MSIS relation as $\begin{bmatrix} \mathbf{b}_1 | 0 \\ \mathbf{0} | g_i \end{bmatrix} \begin{bmatrix} \mathbf{r}_i^\top \\ \mathsf{M} \end{bmatrix}$, we can prove that $t_{1,i}$ has the correct form for some small $(\mathbf{r}'_i, \mathsf{M}'_i)$ using standard NIZKs for MSIS relations. We can then further prove that $\mathsf{M}'_i = \mathsf{M}'_{i+1}$ for all $i \in [k-1]$ by proving linear relations between $t_{1,i}$ and $t_{1,i+1}$.

It turns out that for concrete efficiency, the extraction/soundness slack on $\mathbf{R}$ has a very large impact on the final signature size. For instance, if we use Lyubashevsky's NIZK [36, 37] to prove the MSIS relation, we are only able to extract a witness $(\mathbf{R}', I')$ such that $\widehat{c} \cdot \mathbf{t}_1 = \mathbf{b}_1\mathbf{R}' + I' \cdot \mathbf{g}$ for some small and invertible $\widehat{c}$. Although $\widehat{c}$ is relatively small, this negatively impacts the size of the short vector sampled by the signer, which then negatively impacts the witness size used by the user to construct the final blind signature. Due to the way the slackness propagates in each step, the blow-up in the parameter accumulates and the final blind signature can become quite large.

To this end, we use the exact proof by Bootle et al. [16] to prove the MSIS relation and glue the proof of linear relation together. This allows the reduction to extract an *exact* witness with regards to $\mathbf{R}'$ but a *relaxed* witness with regards to the message I'. This idea is somewhat similar to the very recent "hybrid exact/relaxed" lattice proofs introduced in an independent and concurrent work by Esgin et al. [29]. We finish by showing that we can apply the Katsumata transform to this new protocol to obtain the desired multi-proof straight-line extractable NIZK. Here, we highlight that while using a more complex NIZK has a positive impact on the final blind signature size, it harms the communication cost from the user to the signer. This is because the exact proof of Bootle et al. [16] has a larger proof size compared to the standard NIZK for $\mathsf{MSIS}/\mathsf{MLWE}$ relations. If we wanted to minimize the sum of the communication cost and signature size, then other NIZKs could be a better fit. We believe one of the benefits of our generic construction is that one can choose different instantiations of the NIZKs to optimize the scheme concerning their specific metric. We also note that we were not able to use the more recent efficient exact-proof NIZKs [10, 28] since it was non-trivial to apply the Katsumata transform. We leave it as an interesting open question to extend the Katsumata transform to these efficient NIZKs.

Finally, the above NIZK gives us full straight-line extraction capability but we show that we can relax this when considering the concrete proof of one-more unforgeability of our blind signature (in the classical ROM). This allows us to reduce the proof size of our NIZK by roughly 40 folds (i.e. from 34 MB to 851 KB). At a very high level, the Katsumata transform applied to the proof of the linear relation already allows us to straight-line extract a *relaxed* relation with regards to $\mathbf{R}'$ as well. If $\mathbf{R}'$ is not the same as the $\mathbf{R}''$ extracted from the *exact* relation of the proof of Bootle et al., then it turns out that we can solve the MSIS problem. In other words, unless the adversary against the one-more unforgeability breaks the MSIS assumption, the $\mathbf{R}'$ that the reduction straight-line extracts from the linear relation are exact, rather than being relaxed. Hence, the reduction tries to straight-line extract from the linear proof, and if it fails to extract an exact witness $\mathbf{R}'$, then it can quit the simulation of the one-more unforgeability game.

It then simply resorts to rewinding the adversary to extract $\mathbf{R}''$ from the exact proof of Bootle et al. aiming to break the MSIS problem. Thus, we can reduce the proof size by removing the Katsumata transform applied the exact proof of Bootle et al.

2 Preliminaries

2.1 Blind Signature

We provide the definition of blind signatures. For simplicity, we give a definition focusing on round-optimal (i.e. two-round) blind signatures.

Definition 2.1 (Blind Signature). *A round-optimal blind signature scheme Π_{BS} with a message space $\mathcal{M}$ consists of PPT algorithms $(\mathsf{BSGen}, \mathcal{U}_1, \mathcal{S}_2, \mathcal{U}_{\mathsf{der}}, \mathsf{BSVerify})$ defined as follows:*

$\mathsf{BSGen}(1^\lambda) \to (\mathsf{vk}, \mathsf{sk})$**:** *The key generation algorithm takes as input the security parameter 1^λ and outputs a verification key vk and a signing key sk.*

$\mathcal{U}_1(\mathsf{vk}, \mathsf{M}) \to (\rho_1, \mathsf{st}_{\mathcal{U}})$**:** *This is the user's first message generation algorithm that takes as input a verification key vk and a message $\mathsf{M} \in \mathcal{M}$ and outputs a first message ρ_1 and a state $\mathsf{st}_{\mathcal{U}}$.*

$\mathcal{S}_2(\mathsf{sk}, \rho_1) \to \rho_2$**:** *This is the signer's second message generation algorithm that takes as input a signing key sk and a first message ρ_1 as input and outputs a second message ρ_2.*

$\mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2) \to \Sigma$**:** *This is the user's signature derivation algorithm that takes as input a state $\mathsf{st}_{\mathcal{U}}$ and a second message ρ_2 as input and outputs a signature Σ.*

$\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}, \Sigma) \to \top$ **or** $\bot$**:** *This is a deterministic verification algorithm that takes as input a verification key vk, a message $\mathsf{M} \in \mathcal{M}$, and a signature Σ, and outputs $\top$ to indicate acceptance or $\bot$ to indicate rejection.*

Definition 2.2 (Correctness). *A blind signature is* correct *if for any $\lambda \in \mathbb{N}$ and $\mathsf{M} \in \mathcal{M}$, we have $\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}, \Sigma) = \top$ with overwhelming probability when $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{BSGen}(1^\lambda)$, $(\rho_1, \mathsf{st}_{\mathcal{U}}) \xleftarrow{\$} \mathcal{U}_1(\mathsf{vk}, \mathsf{M})$, $\rho_2 \xleftarrow{\$} \mathcal{S}_2(\mathsf{sk}, \rho_1)$, and $\Sigma \xleftarrow{\$} \mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2)$.*

Definition 2.3 (One-More Unforgeability). *A blind signature is classically (resp. quantumly)* one-more unforgeable *if for any $\mathsf{Q} = \mathsf{poly}(\lambda)$ and PPT (resp. QPT) adversary $\mathcal{A}$ that makes at most Q classical queries, $\mathsf{Adv}^{\mathsf{OMU}}_{\Pi_{\mathsf{BS}}}(\mathcal{A})$ defined as*

$$\Pr\left[\begin{array}{l}(\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{BSGen}(1^\lambda) \\ \{(\mathsf{M}_i, \Sigma_i)\}_{i \in [\mathsf{Q}+1]} \xleftarrow{\$} \mathcal{A}^{\mathcal{S}_2(\mathsf{sk}, \cdot)}(\mathsf{vk})\end{array} : \begin{array}{l}\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}_i, \Sigma_i) = \top \text{ for all } i \in [\mathsf{Q}+1] \\ \wedge \{\mathsf{M}_i\}_{i \in [\mathsf{Q}+1]} \text{ is pairwise distinct}\end{array}\right],$$

is $\mathsf{negl}(\lambda)$, where we say that $\{\mathsf{M}_i\}_{i \in [\mathsf{Q}+1]}$ is pairwise distinct if we have $\mathsf{M}_i \neq \mathsf{M}_j$ for all $i \neq j$.

Definition 2.4 (Blindness Under Malicious Keys). *To define blindness, we consider the following game between an adversary $\mathcal{A}$ and a challenger.*

Setup. *$\mathcal{A}$ is given as input the security parameter 1^λ, and sends a verification key vk and a pair of messages $(\mathsf{M}_0, \mathsf{M}_1)$ to the challenger.*

First Message. *The challenger generates $(\rho_{1,b}, \mathsf{st}_{\mathcal{U},b}) \xleftarrow{\$} \mathcal{U}_1(\mathsf{vk}, \mathsf{M}_b)$ for each $b \in \{0,1\}$, picks $\mathsf{coin} \xleftarrow{\$} \{0,1\}$, and gives $(\rho_{1,\mathsf{coin}}, \rho_{1,1-\mathsf{coin}})$ to $\mathcal{A}$.*

Second Message. *The adversary sends $(\rho_{2,\mathsf{coin}}, \rho_{2,1-\mathsf{coin}})$ to the challenger.*

Signature Derivation. *The challenger generates $\Sigma_b \xleftarrow{\$} \mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U},b}, \rho_{2,b})$ for each $b \in \{0,1\}$. If $\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}_b, \Sigma_b) = \bot$ for either $b = 0$ or 1, then the challenger gives $(\bot, \bot)$ to $\mathcal{A}$. Otherwise, it gives (Σ_0, Σ_1) to $\mathcal{A}$.*

Guess. *$\mathcal{A}$ outputs its guess coin'.*

We say that $\mathcal{A}$ wins if $\mathsf{coin} = \mathsf{coin}'$. We say that a blind signature is classically (resp. quantumly) blind against malicious senders *if for any PPT (resp. QPT) adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{blind}}_{\Pi_{\mathsf{BS}}}(\mathcal{A}) := |\Pr[\mathcal{A} \text{ wins}] - 1/2| = \mathsf{negl}(\lambda)$.*

2.2 Non-Interactive Zero-Knowledge Proofs in the (Q)ROM

We consider a non-interactive zero-knowledge proof of knowledge (or simply NIZK) in the (Q)ROM. We assume that the prover and verifier are provided with a common *random* string crs. Looking ahead, our blind signature generates this crs as the output of another random oracle so it does not rely on any trusted setup, thus making the blind signature also blind against malicious senders.

Definition 2.5 (NIZK Proof System). *A non-interactive zero-knowledge (NIZK) proof system Π_{NIZK} for the relations $\mathcal{R}$ and $\mathcal{R}_{\mathsf{gap}}$ (which are implicitly parameterized by the security parameter λ)[10] and a common random string crs with length $\ell(\lambda)$ consists of oracle-calling PPT algorithms $(\mathsf{Prove}, \mathsf{Verify})$ defined as follows:*

$\mathsf{Prove}^{\mathcal{O}}(\mathsf{crs}, \mathsf{X}, \mathsf{W}) \to \pi/\bot$: *The prover algorithm takes as inputs a common random string $\mathsf{crs} \in \{0,1\}^\ell$, statement and witness pair $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$, and outputs a proof π or a special symbol $\bot$ denoting abort.*

$\mathsf{Verify}^{\mathcal{O}}(\mathsf{crs}, \mathsf{X}, \pi) \to \top/\bot$: *The verifier algorithm takes as inputs a crs, a statement X and a proof π, and outputs either $\top$ (accept) or $\bot$ (reject).*

We denote by $\mathcal{L}_{\mathcal{R}} := \{\mathsf{X} \mid \exists \mathsf{W}, (\mathsf{X}, \mathsf{W}) \in \mathcal{R}\}$ the language induced by $\mathcal{R}$. Moreover, we may omit crs when they are not required.

We rely on the standard notions of correctness, zero-knowledge, and *single-proof extractable* NIZKs, which is typically defined as a specific type of *proof of knowledge* in the literature. Below, we define a strong type of proof of knowledge where we can directly extract from multiple statement and proof pairs output by the adversary.

[10] Unlike conventional definition of "gap" soundness, we do not require $\mathcal{R} \subseteq \mathcal{R}_{\mathsf{gap}}$ to hold. The NIZK is useful as long as $\mathcal{R}_{\mathsf{gap}}$ defines a hard language.

Definition 2.6 (Multi-Proof Extractability). *An* NIZK *proof system* Π_{NIZK} *is classically (resp. quantumly)* multi-proof extractable *if there exists a PPT (resp. QPT) oracle simulator* $\mathcal{S}_{\mathsf{crs}}$ *and a PPT (resp. QPT) extractor* Multi-Extract *with the following properties:*

CRS Indistinguishability. *For any PPT (resp. QPT) adversary* $\mathcal{A}$*, the following advantage* $\mathsf{Adv}^{\mathsf{crs}}_{\Pi_{\mathsf{NIZK}}}(\mathcal{A})$ *is* $\mathsf{negl}(\lambda)$*:*

$$\left|\Pr[\mathsf{crs} \xleftarrow{\$} \{0,1\}^{\ell} : \mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{crs}) = 1] - \Pr[(\widetilde{\mathsf{crs}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^{\lambda}) : \mathcal{A}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{crs}}) = 1]\right|.$$

Straight-Line Extractability. *There exists constants* c, e_1, e_2 *and polynomial* $p(\lambda)$ *such that for any* $\mathsf{Q}_{\mathsf{H}} = \mathsf{poly}(\lambda)$ *and PPT (resp. QPT) adversary* $\mathcal{A}$ *that makes at most* Q_{H} *random oracle queries with*

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^{\lambda}),\\ \{(\mathsf{X}_i, \pi_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}]} \xleftarrow{\$} \mathcal{A}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{crs}})\end{array} : \forall i \in [\mathsf{Q}_{\mathsf{S}}], \mathsf{Verify}^{\mathcal{O}}(\widetilde{\mathsf{crs}}, \mathsf{X}_i, \pi_i) = \top\right] \geq \mu(\lambda),$$

we have,

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^{\lambda}), \{(\mathsf{X}_i, \pi_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}]} \xleftarrow{\$} \mathcal{A}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{crs}}),\\ \{\mathsf{W}_i \xleftarrow{\$} \mathsf{Multi\text{-}Extract}(1^{\lambda}, \mathsf{Q}_{\mathsf{H}}, \mathsf{Q}_{\mathsf{S}}, 1/\mu, \tau, \mathsf{X}_i, \pi_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}]}\end{array} : \begin{array}{l}\forall i \in [\mathsf{Q}_{\mathsf{S}}], (\mathsf{X}_i, \mathsf{W}_i) \in \mathcal{R}_{\mathsf{gap}}\\ \wedge\ \mathsf{Verify}^{\mathcal{O}}(\widetilde{\mathsf{crs}}, \mathsf{X}_i, \pi_i) = \top\end{array}\right]$$

is larger than $\mu(\lambda)/2 - \mathsf{negl}(\lambda)$*. Moreover, the runtime of* Multi-Extract *is upper bounded by* $\mathsf{Q}_{\mathsf{H}}^{e_1} \cdot \mathsf{Q}_{\mathsf{S}}^{e_2} \cdot \frac{1}{\mu^c} \cdot p(\lambda)$*.*

We show that for our NIZK, we have $(c, e_1, e_2) = (1, 1, 0)$ in the classical setting where $p(\lambda)$ is roughly the time it takes to perform a standard PKE decryption. In the quantum setting, we instead have $(c, e_1, e_2) = (1, 2, 1)$.

3 Lattice-based Blind Signature from Compatible Commitments

In this section, we provide our generic construction of a blind signature tailored to lattices. A high level overview of our construction is provided in Sec. 1.3.

3.1 Trapdoor-Sampling-Compatible Commitments

We first explain the type of lattice-based commitments applicable to our generic construction, which we call *trapdoor-sampling-compatible* commitments. For instance, the BDLOP commitment by Baum et al. [12] is one specific instantiation. We keep this layer of abstraction as we believe this captures the essential properties required by our generic construction and allows drop-in of different types of commitments.

Definition 3.1 (Trapdoor-Sampling-Compatibility). *Let* L *and* ℓ_{com} *be positive integers. Let* Π_{Com} *be a commitment scheme with message space*

$\mathcal{M} := R_q^L$ *and an* ℓ_{com}*-bit common random string* crs*.* Π_{Com} *is* (k, δ)-trapdoor-sampling-compatible *if there exists accompanying deterministic PT algorithms* $(\mathsf{ParseCom}, \mathsf{ParseRand})$ *such that for any* $\mathsf{crs} \in \{0,1\}^{\ell_{\mathsf{com}}}$*,* $\mathsf{rand} \in \mathcal{R}$*,* $\mathsf{M} \in \mathcal{M}$*, and* $\mathsf{com} = \mathsf{Com}(\mathsf{crs}, \mathsf{M}; \mathsf{rand})$*, we have the following:*

- $(\mathbf{b}_i)_{i\in[L]} \subseteq \mathsf{crs}$[11]*,* $\mathbf{t} = \mathsf{ParseCom}(\mathsf{com})$*, and* $(\mathbf{r}_i)_{i\in[L]} = \mathsf{ParseRand}(\mathsf{rand})$*, where* $\mathbf{b}_i \in R_q^k, \mathbf{t} \in R_q^L$*, and* $\mathbf{r}_i \in R^k$*;*
- *for each* $i \in [L]$*,* $t_i = \mathbf{b}_i \mathbf{r}_i^\top + \mathsf{M}_i \in R_q$*, where* t_i *is the* i*-th entry of* $\mathbf{t}$*,* M_i *is the* i*-th entry of* $\mathbf{M}$*, and* $\mathbf{r}_i$ *satisfies* $s_1\big([\mathbf{r}_1^\top | \dots | \mathbf{r}_L^\top]\big) \le \delta$*;*
- *finally, the concatenated vector* $[\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_L] \in R_q^{Lk}$ *consists of elements in* $\{0,1\} \subset R_q$ *or uniform random elements in* R_q*, where the probability is taken over the randomness of* $\mathsf{crs} \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{com}}}$*. Note that when* $\mathbf{b}_i$ *and* $\mathbf{b}_j$ *contain duplicate entries, say the first entry of* $\mathbf{b}_i$ *and* $\mathbf{b}_j$ *are defined identically, then we only consider randomness over one of them.*

Roughly, δ dictates the "quality" of the randomness used to hide the message. The choice of the spectral norm $s_1(\cdot)$ is arbitrary, and for instance, we can use the two-norm.

3.2 Construction of Blind Signature

Parameters. For reference, we provide in Table 1 the parameters used in the scheme and in the security proof. The main parameters to keep in mind are (q, d, k_1, k_2, k_3): q and d define the polynomial ring R_q; k_1 is the lattice dimension used to perform trapdoor sampling; k_2 is the dimension of the message space $\mathcal{M}$ of the commitment scheme Π_{Com}; and k_3 is the length of $(\mathbf{b}_i)_{i\in[L=k_2]}$ of Π_{Com}. For those only interested in the asymptotic, one can safely assume k_1, k_2, k_3 are the same value.

Building Blocks. Our blind signature Π_{BS} relies on the following building blocks. The norm bounds on vectors and matrices are chosen with the later concrete parameter selection in mind. For the asymptotic result, we could have simply used the two-norm.

- A commitment scheme Π_{Com} with message space $\mathcal{M} = R_q^{k_2}$ (i.e., $L := k_2$ in Def. 3.1), randomness space $\mathcal{R}$, and an ℓ_{com}-bit common random string $\mathsf{crs}_{\mathsf{com}}$ that satisfies hiding and (k_3, δ)-*trapdoor-sampling-compatiblity.*
- A NIZK proof system $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ (without a common random string) for the relations $\mathcal{R}^{\mathsf{s}}$ and $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$ that satisfies correctness, zero-knowledge and *single-proof* extractability, where $\mathcal{R}^{\mathsf{s}}$ and $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$ are defined as follows:[12]
  - $\mathcal{R}^{\mathsf{s}} := \left\{ \begin{array}{l|c} \mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, & (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}} \in R^{k_1+k_2+k_2\cdot k_3}, \\ \quad (\mathbf{b}_i)_{i\in[k_2]}, u, h), & \forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U}}_{\Sigma,i} \\ \mathsf{W} = \tilde{\mathbf{e}} & \wedge\ [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}]\, \tilde{\mathbf{e}}^\top = u \end{array} \right\};$

[11] That is, we assume the bit-representation of each $\mathbf{b}_i$ is included in crs. Without loss of generality, we can think instead that crs lives in $(R_q^k)^L \times \{0,1\}^\ell$.

[12] With an abuse of notation, when we write $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) = \tilde{\mathbf{e}} \in R^{k_1+k_2+k_2\cdot k_3}$, we assume $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) \in R^{k_1} \times R^{k_2} \times R^{k_2\cdot k_3}$.

Parameter	Explanation
R_q	Polynomial ring $R_q = \mathbb{Z}[X]/(q, X^d + 1)$
B_{inv}	Any $a \in R_q$ s.t. $\|a\|_2 \leq B_{\mathsf{inv}}$ is invertible
k_1	Size of lattice trapdoor $\mathbf{T} \in R^{k_1 \times k_1}$
k_2	Size of the message space $\mathcal{M} = R_q^{k_2}$ for Π_{Com}
(k_3, δ)	Parameters for the trapdoor-sampling-compatible Π_{Com}
σ	Gaussian parameter for trapdoor sampling
$(\ell^{\mathsf{m}}_{\mathsf{NIZK}}, \ell_{\mathsf{com}})$	Length of crs for $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ and Π_{Com}
δ^{gap}	Spectral norm bound on the extracted com. rand.
$B^{\mathcal{S}}_{\Sigma,i}$, $i \in [3]$	Two-norm bound on $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) := \mathbf{e}$ sampled by the signer
$B^{\mathcal{U}}_{\Sigma,i}$, $i \in [3]$	Two-norm bound on real secret $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}}$
$B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i}$, $i \in [3]$	Two-norm bound on extracted $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}}$
$S_{\mathsf{chal}} \subset R_q$	Challenge set of the interactive proof sys. implicit in $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$
B_c	One-norm bound on $c \in S_{\mathsf{chal}}$
$S_{\mathsf{hash}} \subset R_q$	Hashed message set with size $> 2^\lambda$ s.t. $\forall (c,h) \in S_{\mathsf{chal}} \times S_{\mathsf{hash}}$, $\|c \cdot h\|_2 \leq B_{\mathsf{inv}}/2$
Δ_{MLWE}	Bound s.t. *search* MLWE has non-unique solution
$(\chi_{\mathsf{MLWE}}, B_{\mathsf{MLWE}})$	Noise distribution for *decision* MLWE, where $\mathbf{R} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1 \times k_2} \Rightarrow s_1(\mathbf{R}) \leq B_{\mathsf{MLWE}}$ w.o.p
$(\chi_{\mathsf{DSMR}}, B_{\mathsf{DSMR}})$	Noise distribution $\chi_{\mathsf{DSMR}} := D_{\mathbb{Z}, B_{\mathsf{DSMR}}}$ for DSMR
B_{MSIS}	Two-norm bound on the solution for MSIS

Table 1: Overview of parameters and notations. The rows following the second double horizontal line are parameters mainly used in the security proof.

- $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}} := \left\{ \begin{array}{l} \mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, \\ \quad (\mathbf{b}_i)_{i \in [k_2]}, u, h), \\ \mathsf{W} = (\tilde{\mathbf{e}}, c) \end{array} \middle| \begin{array}{c} (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}} \in R^{k_1+k_2+k_2 \cdot k_3}, \\ \forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \leq B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i} \ \wedge \ \|c\|_1 \leq B_c \\ \wedge \ [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}]\, \tilde{\mathbf{e}}^\top = c \cdot u \end{array} \right\}.$

– A NIZK proof system $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ (with a common random string $\mathsf{com}^{\mathsf{m}}_{\mathsf{NIZK}}$) for the relations $\mathcal{R}^{\mathsf{m}}$ and $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ that satisfies correctness, zero-knowledge and *multi-proof* extractability, where $\mathcal{R}^{\mathsf{m}}$ and $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ are defined as follows:

- $\mathcal{R}^{\mathsf{m}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \\ \mathsf{W} = (h, \mathsf{rand}) \end{array} \middle| \begin{array}{c} (h, \mathsf{rand}) \in S_{\mathsf{hash}} \times \mathcal{R}, \\ \wedge\ \mathsf{com} = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, h \cdot \mathbf{g}; \mathsf{rand}) \end{array} \right\};$
- $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \\ \mathsf{W} = (h', c', c, (\mathbf{r}_i)_{i \in [k_2]}) \end{array} \middle| \begin{array}{c} \|h'\|_2 \leq B_{\mathsf{inv}}/2 \ \wedge \ \|c'\|_1, \|c\|_1 \leq B_c \\ \wedge\ s_1\left([\mathbf{r}_1^\top | \cdots | \mathbf{r}_{k_2}^\top]\right) \leq \delta^{\mathsf{gap}} \\ \wedge\ t_i = \mathbf{b}_i(\mathbf{r}_i/c)^\top + (h'/c') \cdot g_i \end{array} \right\},$

  where $\mathbf{t} = \mathsf{ParseCom}(\mathsf{com})$, $(\mathbf{b}_i)_{i \in [k_2]} \subseteq \mathsf{crs}_{\mathsf{com}}$, $\mathbf{g} = [1 \mid b \mid \cdots \mid b^{k_2 - 1}] \in R_q^{k_2}$ is the gadget matrix with $k_2 = \lceil \log_b(q) \rceil$, and g_i is the i-th element of $\mathbf{g}$.

– Four hash functions $\mathsf{H}_{\mathsf{crs}}$, H_{M}, H_{m}, and H_{s} modeled as a random oracle in the security proof. The latter two H_{m} and H_{s} are hash functions used by the NIZK proof systems $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$, respectively. $\mathsf{H}_{\mathsf{M}} : \{0,1\}^* \to R_q$ is a hash function used to map messages to ring elements. $\mathsf{H}_{\mathsf{crs}}$ is a special hash function, for which we only use the input 0. Specifically, $\mathsf{H}_{\mathsf{crs}}(0) =$

$(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2)$ contains the common random strings $\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}$ and $\mathsf{crs}_{\mathsf{com}}$ used by $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ and Π_{Com}, respectively, and a random vector $\mathbf{a}_2 \in R_q^{k_2}$.

Construction. The construction of our blind signature Π_{BS} is provided below. We assume $\mathsf{H}_{\mathsf{crs}}(0) = (\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2)$ and $(\mathbf{b}_i)_{i\in[k_2]} \subseteq \mathsf{crs}_{\mathsf{com}}$ are derived correctly by all the algorithms and omit the process of generating them.

$\mathsf{BSGen}(1^\lambda)$: It runs $(\mathbf{a}_1, \mathbf{T}_{\mathbf{a}_1}) \xleftarrow{\$} \mathsf{TrapGen}(1^{k_1 d}, q)$, samples $\mathbf{s} \xleftarrow{\$} [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]^{(k_1+k_2k_3)}_{\mathsf{coeff}}$[13] and sets $u = [\mathbf{a}_1 \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \cdot \mathbf{s}^\top \in R_q$, where recall $\mathbf{a}_1 \in R_q^{k_1}$, $\mathbf{b}_i \in R_q^{k_3}$ for $i \in [k_2]$. It then outputs $(\mathsf{vk}, \mathsf{sk}) = ((\mathbf{a}_1, u), \mathbf{T}_{\mathbf{a}_1})$.

$\mathcal{U}_1(\mathsf{vk}, \mathsf{M})$: It hashes $h = \mathsf{H}_{\mathsf{M}}(\mathsf{M})$, samples $\mathsf{rand} \xleftarrow{\$} \mathcal{R}$, and computes $\mathsf{com} = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, h \cdot \mathbf{g}; \mathsf{rand})$. It then creates a proof $\pi^{\mathsf{m}} \xleftarrow{\$} \mathsf{Prove}^{\mathsf{H}_{\mathsf{m}}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), (h, \mathsf{rand}))$ that proves the wellformedness of the commitment com, and outputs the first message $\rho_1 = (\mathsf{com}, \pi^{\mathsf{m}})$. Finally, it sets its state as $\mathsf{st}_{\mathcal{U}} = \mathsf{rand}$.

$\mathcal{S}_2(\mathsf{sk}, \rho_1)$: It parses $(\mathsf{com}, \pi^{\mathsf{m}}) \leftarrow \rho_1$ and outputs $\perp$ if $\mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \pi^{\mathsf{m}}) = \perp$. Otherwise, it computes $\mathbf{t} \leftarrow \mathsf{ParseCom}(\mathsf{com})$ and samples a short vector $\mathbf{e} \in R^{k_1+k_2+k_2k_3}$ such that

$$[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \cdot \mathbf{e}^\top = u, \tag{6}$$

using $\mathbf{e} \xleftarrow{\$} \mathsf{SampleLeft}(\mathbf{a}_1, [\mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}], u, \mathbf{T}_{\mathbf{a}_1}, \sigma)$. It outputs the second message $\rho_2 = \mathbf{e}$.

$\mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2)$: It parses $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) := \mathbf{e} \leftarrow \rho_2$, $\mathsf{rand} \leftarrow \mathsf{st}_{\mathcal{U}}$, and outputs $\perp$ if either $\exists i \in [3], \|\mathbf{e}_i\|_2 > B^{\mathcal{S}}_{\Sigma,i}$ or Eq. (6) does not hold. Otherwise, it computes $\mathbf{t} \leftarrow \mathsf{ParseCom}(\mathsf{com}_{\mathsf{crs}})$ and $(\mathbf{r}_i)_{i\in[k_2]} \leftarrow \mathsf{ParseRand}(\mathsf{rand})$, where $h = \mathsf{H}_{\mathsf{M}}(\mathsf{M})$, $t_i = \mathbf{b}_i \mathbf{r}_i^\top + h \cdot g_i \in R_q$, and t_i and g_i are the i-th entries of $\mathbf{t}$ and $\mathbf{g}$, respectively. It then rewrites the left hand side of Eq. (6) as follows:

$$\begin{aligned}
&[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \cdot \mathbf{e}^\top \\
&= \left[\mathbf{a}_1 \mid \mathbf{a}_2 + [\mathbf{b}_1\mathbf{r}_1^\top + h \cdot g_1 \mid \cdots \mid \mathbf{b}_{k_2}\mathbf{r}_{k_2}^\top + h \cdot g_{k_2}] \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}\right] \cdot \mathbf{e}^\top \\
&= [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \underbrace{\begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \\ e_{2,1} \cdot \mathbf{r}_1^\top + \mathbf{e}_{3,1}^\top \\ \cdots \\ e_{2,k_2} \cdot \mathbf{r}_{k_2}^\top + \mathbf{e}_{3,k_2}^\top \end{bmatrix}}_{=:\tilde{\mathbf{e}} \in R^{k_1+k_2+k_2k_3}},
\end{aligned}$$

where $\mathbf{e}_3 = [\mathbf{e}_{3,1} \mid \cdots \mid \mathbf{e}_{3,k_2}] \in R^{k_2k_3}$ and $\mathbf{e}_2 = [e_{2,1} \mid \cdots \mid e_{2,k_2}] \in R^{k_2}$ are parsed into appropriate sizes. It then creates a proof $\pi^{\mathsf{s}} \xleftarrow{\$} \mathsf{Prove}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h), \tilde{\mathbf{e}})$ that proves knowledge of a short vector $\tilde{\mathbf{e}}$. If $\perp \leftarrow \mathsf{Verify}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h), \pi^{\mathsf{s}})$, then it outputs $\Sigma = \perp$. Otherwise, it outputs $\Sigma = \pi^{\mathsf{s}}$ as the signature.

[13] For integers a and b such that $a < b$, $[a, b]_{\mathsf{coeff}} \subset R_q$ denotes the set of all polynomials in R_q with coefficients in $[a, b]$.

$\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}, \Sigma)$: It parses $\pi^{\mathsf{s}} \leftarrow \Sigma$, sets $h = \mathsf{H}_{\mathsf{M}}(\mathsf{M})$, and returns the output of $\mathsf{Verify}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h), \pi^{\mathsf{s}})$.

Remark 3.1 (Variations of the Construction). We can consider slight variations of the above construction. For instance, in case the commitment vectors satisfy $\mathbf{b}_1 = \cdots = \mathbf{b}_{k_2}$, which is the case for our concrete instantiation in Sec. 4.1, the signer can alternatively sample $\mathbf{e}$ such that $[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1] \cdot \mathbf{e}^\top = u$ instead of Eq. (6). Which variation offers the "best" blind signature highly depends on many factors: the criteria that we wish to optimize (e.g., minimize the signature size, minimize the total communication cost); the concrete choice of NIZKs and commitments we use; and other implicit parameter selections.

The proof of correctness consists of a routine check. Blindness under malicious keys follows from a standard proof using the zero-knowledge and hiding of the underlying NIZKs and commitment.

3.3 Proof of One-More Unforgeability

The following establishes that our blind signature is one-more unforgeable even against quantum adversaries in the QROM.

Theorem 3.1. *The blind signature Π_{BS} is quantumly one-more unforgeable if the two* NIZK*s $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ for $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ and $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ for $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$ are quantumly single-proof and multi-proof extractable, respectively, and the $\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}$, $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$, $\mathsf{DSMR}_{d,k_1-1,\chi_{\mathsf{DSMR}},q,1}$ and $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ problems are hard.*

Proof Sketch. Assume there exists a QPT adversary $\mathcal{A}$ with non-negligible advantage ϵ against the one-more unforgeability game that makes at most Q_{S} (classical) signature queries. Further assume $\mathcal{A}$ makes at most $\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}$ (resp. $\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}$, $\mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}$, $\mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}$) (quantum) random oracle queries to H_{M} (resp. $\mathsf{H}_{\mathsf{crs}}$, H_{m}, H_{s}). We consider a sequence of games, where we denote E_i as the event $\mathcal{A}$ wins in Game_i. Game_1 is the real one-more unforgeability game.

Game_2 : The challenger simulates all the QRO's by using $2\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}$-wise independent hash functions. This allows the challenger to *efficiently* simulate the QROs.

Game_3 : The challenger programs $\mathsf{H}_{\mathsf{crs}}(0)$ to use the simulated CRS $\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}$ output by the CRS simulator $\mathcal{S}_{\mathsf{crs}}$ of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$.

Game_4 : When $\mathcal{A}$ submits $\rho_{j,1} = (\mathsf{com}_j, \pi^{\mathsf{m}}_j)$ to the challenger as its j-th ($j \in [\mathsf{Q}_{\mathsf{S}}]$) first message, the challenger runs $\mathsf{W}_j \leftarrow \mathsf{Multi\text{-}Extract}(1^\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}, \mathsf{Q}_{\mathsf{S}}, 1/\mu, \tau, \mathsf{X}_j, \pi^{\mathsf{m}}_j)$, where $\mu = \Pr[\mathsf{E}_3]$ and $\mathsf{X}_j = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}_j)$. Due to the definition of the multi-proof extractor $\mathsf{Multi\text{-}Extract}$ (see Def. 2.6), the challenger succeeds in extracting a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ with non-negligible probability and runs in time proportional to $\mathsf{Q}^{e_1}_{\mathsf{H}_{\mathsf{m}}} \cdot \mathsf{Q}^{e_2+1}_{\mathsf{S}} \cdot \frac{1}{\mu^c} \cdot p(\lambda)$, which is a polynomial.

Game_5 : The challenger replaces the function $\mathsf{H}_\mathsf{M} : \mathcal{M} \to S_\mathsf{hash} \subset R_q$ by a *small-range distribution*. Specifically, it sets $r = 2 \cdot C_0 \cdot \mathsf{Q}_{\mathsf{H}_\mathsf{M}}^3 / \mu'$, where $\mu' = \Pr[\mathsf{E}_4]$ and C_0 is some universal constant. It then samples $\mathbf{h} = (h_1, \cdots, h_r) \xleftarrow{\$} (S_\mathsf{hash})^r$ and $P \xleftarrow{\$} \mathsf{Func}(\mathcal{M}, [r])$, and defines H_M as $\mathsf{H}_\mathsf{M}(x) = h_{P(x)}$.

Game_6 : The challenger samples a uniformly random index $j^* \xleftarrow{\$} [r]$ at the beginning of the game and performs two types of checks. First, when the challenger extracts $\mathsf{W}_j = (h'_j, c'_j, c_j, (\mathbf{r}_{j,i})_{i\in[k_2]}) \in \mathcal{R}_\mathsf{gap}^\mathsf{m}$ from the first message $\rho_{j,1}$ submitted to by $\mathcal{A}$, the challenger checks if $h'_j/c'_j \neq h_{j^*}$. Moreover, at the end of the game, when $\mathcal{A}$ outputs the forgery $\{(\mathsf{M}_i, \Sigma_i)\}_{i\in[\mathsf{Q}_\mathsf{S}+1]}$, the challenger checks if $\mathsf{M}'_{j^*} \in \{\mathsf{M}_i\}_{i\in[\mathsf{Q}_\mathsf{S}+1]}$ and if $\{\mathsf{H}_\mathsf{M}(\mathsf{M}_i)\}_{i\in[\mathsf{Q}_\mathsf{S}+1]}$ are pairwise distinct.

Game_7 : After it samples $j^* \xleftarrow{\$} [r]$ at the beginning of the game, the challenger sets $\mathbf{a}_2 = \widetilde{\mathbf{a}}_2 - h_{j^*} \cdot \mathbf{g}$ where $\widetilde{\mathbf{a}}_2 \xleftarrow{\$} R_q^k$, and programs $\mathsf{H}_\mathsf{crs}(0)$ to use this $\mathbf{a}_2$.

Game_8 : The challenger gets rid of the trapdoor $\mathbf{T}_{\mathbf{a}_1}$ included in the secret key sk. In particular, the challenger samples $\mathbf{a}_1 \xleftarrow{\$} R_q^{k_1}$, $\mathbf{R} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1 \times k_2}$, and sets $\widetilde{\mathbf{a}}_2 = \mathbf{a}_1\mathbf{R}$. On input the first message $\rho_1 = (\mathsf{com}, \pi^\mathsf{m})$ from $\mathcal{A}$, it extracts $\mathsf{W} = (h', c', c, (\mathbf{r}_i)_{i\in[k_2]}) \in \mathcal{R}_\mathsf{gap}^\mathsf{m}$ and computes

$$
\begin{aligned}
&[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \\
&= \left[\mathbf{a}_1 \mid \mathbf{a}_1\mathbf{R} - h_{j^*} \cdot \mathbf{g} + \left[\frac{\mathbf{b}_1\mathbf{r}_1^\top}{c} + \frac{h'}{c'} \cdot g_1 \mid \cdots \mid \frac{\mathbf{b}_{k_2}\mathbf{r}_{k_2}^\top}{c} + \frac{h'}{c'} \cdot g_{k_2}\right] \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}\right] \\
&= \left[\mathbf{a}_1 \mid \widehat{\mathbf{b}} \mid \left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right] \mathbf{R}' + \left(\frac{h'}{c'} - h_{j^*}\right) \cdot \mathbf{g}\right] \cdot \mathbf{P}_\mathsf{perm},
\end{aligned}
$$

where $\widehat{\mathbf{b}} = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \in R_q^{k_2 k_3}$, $\widehat{\mathbf{R}} = \mathbf{I}_{k_2} \otimes [\mathbf{r}_1^\top \mid \cdots \mid \mathbf{r}_{k_2}^\top] \in R^{k_2 k_3 \times k_2}$, $\mathbf{R}' = \begin{bmatrix} \mathbf{R} \\ \frac{1}{c}\widehat{\mathbf{R}} \end{bmatrix} \in R^{k_2(k_3+1)\times k_2}$, and $\mathbf{P}_\mathsf{perm}$ is a permutation matrix that appropriately reorders the columns. It then samples a short vector $\mathbf{e}' \in R^{k_1+k_2+k_2k_3}$ such that $\left[\mathbf{a}_1 \mid \widehat{\mathbf{b}} \mid \left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right] \mathbf{R}' + \left(\frac{h'}{c'} - h_{j^*}\right) \cdot \mathbf{g}\right] \cdot \mathbf{e}'^\top = u$, using the algorithm $\mathsf{SampleRight}$. By setting the parameters correctly, we have invertibility of $h'/c' - h_{j^*}$ as required by the sampling algorithm. The signer algorithm $\mathcal{S}_2$ finally outputs the second message $\rho_2 = \mathbf{e}'(\mathbf{P}_\mathsf{perm}^{-1})^\top$.

At this point, the challenger in Game_8 no longer relies on a trapdoor for $\mathbf{a}_1$. Using the single-proof extractability of $\Pi_{\mathsf{NIZK}}^\mathsf{s}$, the challenger will be able to extract an MSIS solution with respect to $[\mathbf{a}_1|\widehat{\mathbf{b}}]$. □

3.4 Extension: Partially Blind Signatures

We are able to obtain a *partially* blind signature [2] with a simple modification to our blind signature without increasing the signature size. To bind the signature to a specific common message γ, the signer shifts the public syndrome $u \in R_q$ to $u - \mathsf{H}_{\mathsf{M}c}(\gamma)$, where $\mathsf{H}_{\mathsf{M}c}$ is a newly introduced hash function that is modeled as a random oracle in the security proof.

4 Instantiating Our Generic Construction

In this section, we instantiate our generic construction of blind signature, which in particular involves concretizing the building blocks laid out in Sec. 3.2: the trapdoor-sampling-compatible commitment scheme Π_{Com}, the single-proof extractable NIZK proof system $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$, and the multi-proof extractable NIZK proof system $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$. In Sec. 4.3 we provide a concrete set of parameters for our resulting blind signature scheme.

4.1 Concrete Choices for Trapdoor-Sampling-Compatible Commitments and Single-Proof Extractable NIZK

For the trapdoor-sampling-compatible commitment, we rely on (a slight variant of) the BDLOP commitment by Baum et al. [12]. The common random string is of the form $\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1) := \left([1|\mathbf{b}'_0], [0|1|\mathbf{b}'_1] \right) \in R_{q'}^{k_3} \times R_q^{k_3}$, where we use two different moduli q' and q, and q is the modulus that explicitly showed up in the blind signature construction in the previous section. The commitment to a message $\mathbf{M} = (\mathsf{M}_1, \cdots, \mathsf{M}_L) \in R_q^L$ is

$$\mathsf{com} := [\mathbf{t}_1 // \mathbf{t}_2] = \left(\begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \mathbf{R} + \begin{bmatrix} \mathbf{0} \\ \mathsf{M}_1 \mid \cdots \mid \mathsf{M}_L \end{bmatrix} \begin{matrix} \mod q' \\ \mod q \end{matrix} \right) \in R_{q'}^L \times R_q^L.$$

The single-proof extractable NIZK is based on the basic Lyubashevsky's sigma protocol [36, 37], where soundness is argued through rewinding (or the forking lemma [44, 13] to be precise). One minor difference is that we take advantage of the fact that the witness vector $\tilde{\mathbf{e}} \in R^{k_1+k_2+k_3}$ has unbalanced size; the first $(k_1 + k_2)$-entries are smaller than the last k_3 entries.

4.2 Concrete Choice for Multi-Proof Extractable NIZK

Preparation. Let us prepare some notations. Let $R_{q'} = \mathbb{Z}_{q'}[X]/(X^d + 1)$ be a ring that fully splits and consider the NTT over the ring $R_{q'}$ with $\mathsf{NTT} : R_{q'} \to (\mathbb{Z}_{q'}^d)^\top$, and $\mathsf{NTT}^{-1} : (\mathbb{Z}_{q'}^d)^\top \to R_{q'}$. Here, we make it explicit that NTT and NTT^{-1} operates over column vectors. These notions extend naturally to matrices over $R_{q'}$, where NTT^{-1} is only well-defined when the column length of the matrix is divisible by d. We define $\Phi : R_{q'} \mapsto (\mathbb{Z}_{q'}^d)^\top$ to be the map that sends a polynomial to its (column) coefficient vector. We define $\mathsf{Rot} : R_{q'} \mapsto \mathbb{Z}_{q'}^{d \times d}$ to be the map that sends a polynomial $a \in R_{q'}$ to a matrix whose i-th column is $\Phi\left(a \cdot X^i \mod (X^d + 1)\right)$. It can be checked that for $a, b \in R_{q'}$, we have $\mathsf{Rot}(a)\Phi(b) = \Phi(a \cdot b)$. We extend the definition of Rot to vectors in $R_{q'}$, where we have $\mathsf{Rot}(\mathbf{b})\Phi(a) = \Phi(a \cdot \mathbf{b})$ for $(a, \mathbf{b}) \in R_{q'} \times R_{q'}^n$. Here, note that $\mathsf{Rot}(\mathbf{b}) \in \mathbb{Z}_{q'}^{dn \times d}$ and $\Phi(a) \in {\mathbb{Z}_{q'}^d}^\top$. We use $\circ$ for the component-wise product of matrices over $R_{q'}$. Finally, we define the matrix $\Delta \in R_q^{L \times L}$ such that the first column of Δ is $\mathbf{g}$ and all the diagonal entries except for the $(1,1)$-th entry is -1. Specifically, Δ is invertible over R_q and we have $\mathbf{g}\Delta = [1|0|\cdots|0]$.

Construction. We consider the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$ defined as follows:

$$- \mathcal{R}^{\mathsf{m}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com}), \\ \mathsf{W} = (h, \mathsf{rand} := \mathbf{R}) \end{array} \middle| \begin{array}{c} h \in S_{\mathsf{hash}} \wedge \mathbf{R} \in [-1,1]^{k_3 \times L}_{\mathsf{coeff}}, \\ \wedge\ \mathsf{com} = \left(\begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \mathbf{R} + \begin{bmatrix} \mathbf{0} \\ h \cdot \mathbf{g} \end{bmatrix} \begin{array}{l} \bmod q' \\ \bmod q \end{array} \right) \end{array} \right\};$$

$$- \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com}), \\ \mathsf{W} = (h', c, (\mathbf{r}_i)_{i \in [L]}) \end{array} \middle| \begin{array}{c} \|h'\|_2 \le B_{\mathsf{inv}}/2 \ \wedge\ \|c\|_1 \le B_c \\ \wedge\ \mathbf{t} = \mathsf{ParseCom}(\mathsf{com}) \\ \wedge \mathbf{R} \in [-1,1]^{k_3 \times L}_{\mathsf{coeff}} \\ \wedge\ \forall i \in [L], t_i = \mathbf{b}_1 \mathbf{r}_i^\top + (h'/c) \cdot q^{\frac{i-1}{L}} \end{array} \right\},$$

Notice the gap relation $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ has no slack for the commitment randomness. We recover $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ in Sec. 3.2 by setting $\delta^{\mathsf{gap}} = \sqrt{k_3 L} \cdot d$.

The prove and verify algorithms of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$ are provided in Figs. 1 and 2, respectively. The texts in gray are used by the exact proof of [16], the texts in black without highlight are used to prove linear relations, and finally the texts highlighted in gray are used for multi-proof straight-line extractability as in [34]. The crs for $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ consists of a random element H (used for extraction) and random matrices $(\mathbf{a}_0, (\mathbf{A}_k)_{k \in [4]})$ (used for committing), and the crs for Π_{Com} is a random tuple $(\mathbf{b}_0, \mathbf{b}_1)$. Following prior conventions [16, 34], we prove that $\mathbf{R} \in [0,2]^{k_3 \times L}_{\mathsf{coeff}}$ instead, i.e., $\mathbf{R}$ consists of $\{0,1,2\}$-coefficient polynomials. This is without loss of generality since we can add the all one matrix $\mathbb{1}$ to any $\mathbf{R} \in [-1,1]^{k_3 \times L}_{\mathsf{coeff}}$ to obtain a matrix in $[0,2]^{k_3 \times L}_{\mathsf{coeff}}$.

The protocol uses three polynomial rings: $R_{q'} = \mathbb{Z}_{q'}[X]/(X^d + 1)$ is a fully splitting ring used by Bootle et al's [16] exact proof; $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ is a ring where any small element is invertible and is used by the linear proof; $R_Q = \mathbb{Z}_Q[X]/(X^d + 1)$ is used by the the multi-proof straight-line extractability as in [34], and in particular, we require the NTRU assumption to hold over this ring. The interactive protocol implicit in our NIZK is defined with respect to two challenge spaces. The challenge space used in the second (resp. fourth) flow is $\mathbb{Z}^{\tau}_{q'}$ (resp. $C_X^{\tau\tau'} \times C_{\mathsf{ham}}$, where $C_X := \{X^i \mid i \in [2d]\}$ and C_{ham} is the set of $\{0,1\}$-coefficient polynomials in R_q with Hamming weight smaller than B_c). Specifically, we require any element with two-norm smaller than $2B_c$ to be invertible over R_q. Here, τ and τ' are set so that $q^{\tau} \approx (2d)^{\tau\tau'} \approx 2^{128}$ or asymptotically $1/q^{\tau} \approx 1/(2d)^{\tau\tau'} = \mathsf{negl}(\lambda)$. Our protocol also relies on several different Gaussian distributions. They are used either to perform rejection sampling or to invoke the MLWE and DSMR assumptions. The concrete parameter selection is provided in Sec. 4.3.

Security. Below, we provide the proof sketch of the *classical* multi-proof extractability.

Theorem 4.1. *The* NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ *in Figs. 1 and 2 is classically multi-proof extractable with* $(c_1, e_1, e_2) = (1, 1, 0)$ *and* $p(\lambda) = \mathsf{poly}(\lambda)$ *if the* $\mathsf{DSMR}_{d,1,\chi_{\mathsf{DSMR}},Q,p}$, $\mathsf{MSIS}_{d,1,k_4,16B_{\mathbf{Z}},q'}$, *and* $\mathsf{MSIS}_{d,1,k_3,2(B_{\mathbf{Z}'}+B_c\delta^{\mathsf{gap}}),q'}$ *problems are hard.*

Proof. CRS indistinguishability is a simple consequence of the $\mathsf{DSMR}_{d,1,\chi_{\mathsf{DSMR}},Q,p}$ assumption. The proof of straight-line extractability, which is the most technical proof of this work, consists of three parts. We first show in Lemma 4.1 that (roughly) if the adversary $\mathcal{A}$ outputs a valid proof, then $\mathcal{A}$ must have been able

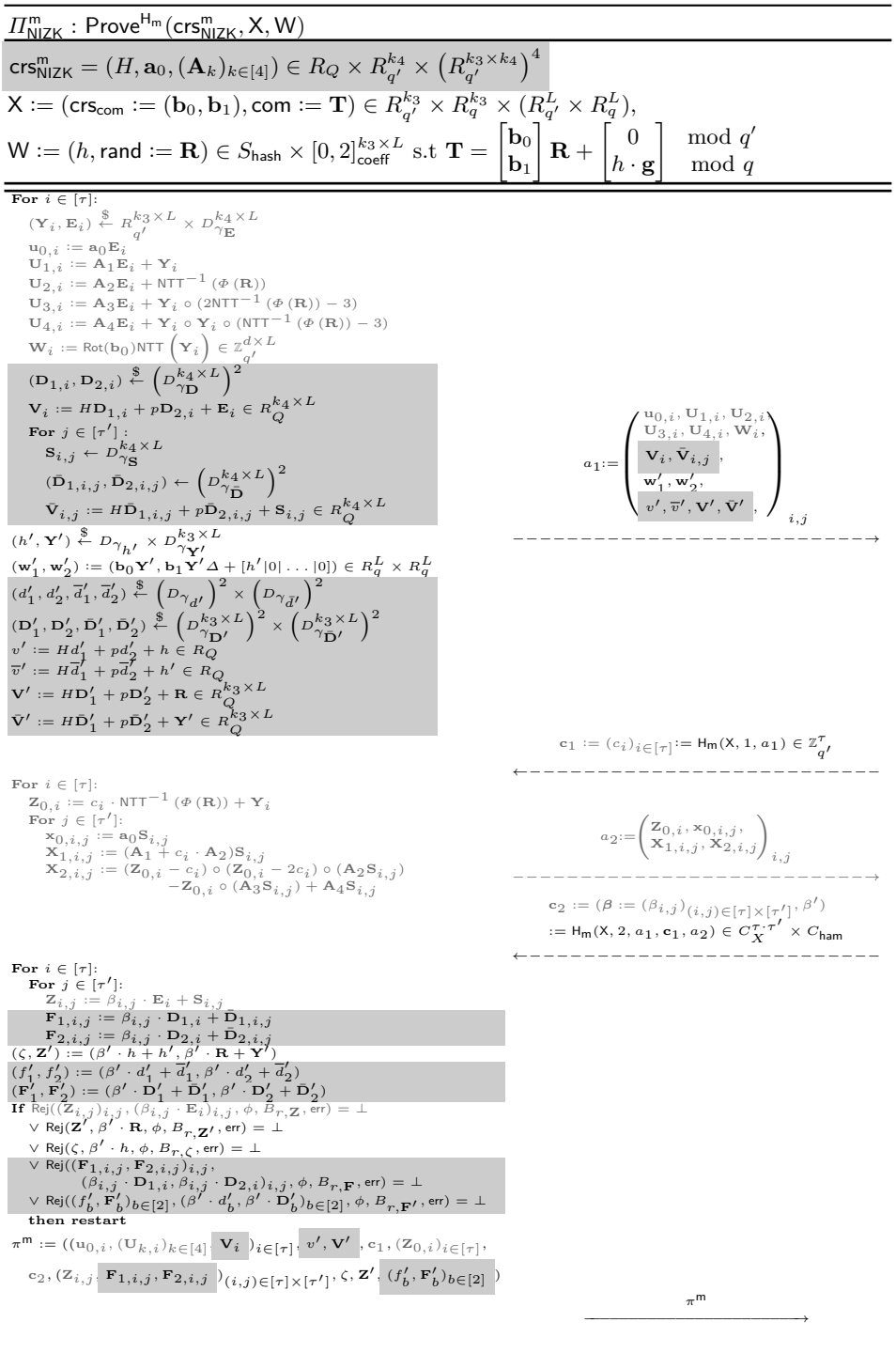

Fig. 1: Prove algorithm for the multi-proof NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$. We illustrate the 5-round interactive protocol that implicitly underlies the NIZK.

$\Pi^{\mathsf{m}}_{\mathsf{NIZK}} : \mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \pi^{\mathsf{m}})$

$\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}} = (H, \mathbf{a}_0, (\mathbf{A}_k)_{k\in[4]}) \in R_Q \times R_{q'}^{k_4} \times \left(R_{q'}^{k_3\times k_4}\right)^4$

$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T}) \in R_{q'}^{k_3} \times R_q^{k_3} \times (R_{q'}^L \times R_q^L)$,

$\pi^{\mathsf{m}} := ((\mathbf{u}_{0,i}, (\mathbf{U}_{k,i})_{k\in[4]}, \mathbf{V}_i)_{i\in[\tau]}, v', \mathbf{V}', \mathbf{c}_1,$

$(\mathbf{Z}_{0,i})_{i\in[\tau]}, \mathbf{c}_2, (\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, (f'_b, \mathbf{F}'_b)_{b\in[2]})$

$\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix} := \mathbf{T} \in R_{q'}^L \times R_q^L$

For $i \in [\tau]$:

$\mathbf{W}_i := \mathsf{Rot}(\mathbf{b}_0)\mathsf{NTT}(\mathbf{Z}_{0,i}) - c_i \cdot \Phi(\mathbf{t}_1) \in \mathbb{Z}_{q'}^{d\times L}$

For $j \in [\tau']$:

$\bar{\mathbf{V}}_{i,j} := H\mathbf{F}_{1,i,j} + p\mathbf{F}_{2,i.j} + \mathbf{Z}_{i,j} - \beta_{i,j} \cdot \mathbf{V}_i \in R_Q^{k_4\times L}$

$\mathbf{x}_{0,i,j} := \mathbf{a}_0\mathbf{Z}_{i,j} - \beta_{i,j} \cdot \mathbf{u}_{0,i} \in R_{q'}^L$

$\mathbf{X}_{1,i,j} := (\mathbf{A}_1 + c_i \cdot \mathbf{A}_2)\mathbf{Z}_{i,j} + \beta_{i,j} \cdot \left(\mathbf{Z}_{0,i} - (\mathbf{U}_{1,i} + c_i \cdot \mathbf{U}_{2,i})\right) \in R_{q'}^{k_4\times L}$

$\mathbf{X}_{2,i,j} := (\mathbf{Z}_{0,i} - c_i) \circ (\mathbf{Z}_{0,i} - 2c_i) \circ (\mathbf{A}_2\mathbf{Z}_{i,j}) - \mathbf{Z}_{0,i} \circ (\mathbf{A}_3\mathbf{Z}_{i,j}) + \mathbf{A}_4\mathbf{Z}_{i,j}$
$-\beta_{i,j} \cdot \left((\mathbf{Z}_{0,i} - c_i) \circ (\mathbf{Z}_{0,i} - 2c_i) \circ \mathbf{U}_{2,i} - \mathbf{Z}_{0,i} \circ \mathbf{U}_{3,i} + \mathbf{U}_{4,i}\right) \in R_{q'}^{k_4\times L}$

$\mathbf{w}'_1 := \mathbf{b}_0\mathbf{Z}' - \beta' \cdot \mathbf{t}_1 \in \mathbb{Z}_{q'}^L$

$\mathbf{w}'_2 := \mathbf{b}_1\mathbf{Z}'\Delta + [\zeta|0|\dots|0] - \beta' \cdot \mathbf{t}_2\Delta \in \mathbb{Z}_{q'}^L$

$\overline{v}' := Hf'_1 + pf'_2 + \zeta - \beta' \cdot v' \in R_Q$

$\bar{\mathbf{V}}' := H\mathbf{F}'_1 + p\mathbf{F}'_2 + \mathbf{Z}' - \beta' \cdot \mathbf{V}' \in R_Q^{k_3\times L}$

$a_1 := \left((\mathbf{u}_{0,i}, \mathbf{U}_{1,i}, \mathbf{U}_{2,i}, \mathbf{U}_{3,i}, \mathbf{U}_{4,i}, \mathbf{W}_i, \mathbf{V}_i, (\bar{\mathbf{V}}_{i,j})_{j\in[\tau']})_{i\in[\tau]}, \mathbf{w}'_1, \mathbf{w}_2, v', \overline{v}', \mathbf{V}', \bar{\mathbf{V}}'\right)$

$a_2 := \left(\mathbf{Z}_{0,i}, (\mathbf{x}_{0,i,j}, \mathbf{X}_{1,i,j}, \mathbf{X}_{2,i,j})_{j\in\times[\tau']}\right)_{i\in[\tau]}$

If
- $\|\zeta\|_2 \geq B$
- $\vee\ \|\mathbf{Z}'\|_2 \geq B_{\mathbf{Z}'}$
- $\vee\ \exists(i,j) \in [\tau]\times[\tau'], \|\mathbf{Z}_{i,j}\|_2 \geq B_{\mathbf{Z}}$
- $\vee\ \|\mathbf{F}'_1\|_\infty \geq B_{1,\mathbf{F}'}$
- $\vee\ \|\mathbf{F}'_2\|_\infty \geq B_{2,\mathbf{F}'}$

then return $\bot$
- $\vee\ \exists(i,j) \in [\tau]\times[\tau'], \|\mathbf{F}_{1,i,j}\|_\infty \geq B_{1,\mathbf{F}}$
- $\vee\ \exists(i,j) \in [\tau]\times[\tau'], \|\mathbf{F}_{2,i,j}\|_\infty \geq B_{2,\mathbf{F}}$
- $\vee\ \mathbf{c}_1 \neq \mathsf{H}_{\mathsf{m}}(\mathsf{X}, 1, a_1)$
- $\vee\ \mathbf{c}_2 \neq \mathsf{H}_{\mathsf{m}}(\mathsf{X}, 2, a_1, \mathbf{c}_1, a_2)$

return $\top$

Fig. 2: Verify algorithm for the multi-proof NIZK for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$.

to succeed on many challenges. That is, the probability that $\mathcal{A}$ succeeds in forging a proof without a witness by guessing the output of the random oracle is at most $\frac{\mu}{2} - \mathsf{negl}(\lambda)$, where μ is the advantage of $\mathcal{A}$ outputting a valid proof. We then show in Lemma 4.2 a specific form of special soundness where an extractor $\mathsf{Extract}_{\mathsf{ss}}$ given the purported proof output by $\mathcal{A}$ along with several specific challenges, extracts a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. We finally provide the description of our straight-line extractor Multi-Extract that internally runs $\mathsf{Extract}_{\mathsf{ss}}$ and bound its success probability.

We present our first lemma which shows that if $\mathcal{A}$ outputs a valid proof, then there must have been multiple challenges for which it could have succeeded on. Formally, we define the sets $\{\Gamma_{1,i}\}_{i\in[\tau]}$ and Γ_2 that count for how many challenges there exists a valid response, and argue that they cannot be too small. More specifically, $\Gamma_{1,i}$ counts the number of second flow challenges c_i for which there exists at least two distinct $\beta_{i,j}$'s included in the fourth flow challenge with a corresponding valid response. Γ_2 on the other hand counts the number of β' included in the fourth flow challenge with a corresponding valid response. Roughly, the former (resp. latter) set is the set of challenges for which $\mathcal{A}$ was able to complete the exact proof of Bootle et al. (resp. proof of linear relation).

Lemma 4.1. *Consider an interactive protocol as defined implicitly in Fig. 1. That is, the transcript is* $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$*, where* $\mathbf{c}_1, \mathbf{c}_2$ *are the challenges the (honest) verifier samples uniformly at random and* resp *is the response* $\left((\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\right)$ *sent by the prover. For any statement* X*, first, second, third, and fourth flows* $a_1, \mathbf{c}_1$*,* a_2*, and* $\mathbf{c}_2$*, respectively, we define the following sets for all* $i \in [\tau]$*:*

$$\Gamma_{1,i}(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2)$$

$$:= \left\{ \bar{c}_i \in \mathbb{Z}_{q'} \;\middle|\; \begin{array}{l} (c_{i'})_{i'\in[\tau]} \leftarrow \mathbf{c}_1, \overline{\mathbf{c}}_1 := (\bar{c}_i) \cup (c_{i'})_{i'\in[\tau]\setminus\{i\}}, \\ (\boldsymbol{\beta} = (\beta_{i',j'})_{(i',j')\in[\tau]\times[\tau']}, \beta') \leftarrow \mathbf{c}_2 \\ \exists j \in [\tau'], \textit{distinct } \left(\overline{\beta}_{i,j}, \overline{\beta}'_{i,j}\right) \in (C_X)^2, \\ \overline{\boldsymbol{\beta}} := (\overline{\beta}_{i,j}) \cup \left(\beta_{i',j'}\right)_{(i',j')\neq(i,j)}, \overline{\boldsymbol{\beta}}' := (\overline{\beta}'_{i,j}) \cup \left(\beta_{i',j'}\right)_{(i',j')\neq(i,j)}, \\ \exists(\overline{a}_2, \overline{a}'_2), (\overline{\mathsf{resp}}, \overline{\mathsf{resp}}') \textit{ s.t. } (a_1, \overline{\mathbf{c}}_1, \overline{a}_2, \mathbf{c}_2 := \left(\overline{\boldsymbol{\beta}}, \beta'\right), \overline{\mathsf{resp}}) \textit{ and} \\ (a_1, \overline{\mathbf{c}}_1, \overline{a}'_2, \overline{\mathbf{c}}_2 := \left(\overline{\boldsymbol{\beta}}', \beta'\right), \overline{\mathsf{resp}}') \textit{ are valid} \end{array} \right\}$$

$$\Gamma_2(\mathsf{X}, a_1, \mathbf{c}_1, a_2, \mathbf{c}_2)$$

$$:= \left\{ \overline{\beta}' \in C_{\mathsf{ham}} \mid (\boldsymbol{\beta}, \beta') \leftarrow \mathbf{c}_2, \exists \overline{\mathsf{resp}} \textit{ s.t. } (a_1, \mathbf{c}_1, a_2, \mathbf{c}_2 := (\boldsymbol{\beta}, \overline{\beta}'), \overline{\mathsf{resp}}) \textit{ is valid} \right\},$$

where we say a transcript $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$ *is* valid *if the proof* π^{m} *implicitly defined by* $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$ *is valid for statement* X*.*

Then, for any $\mathsf{Q}_{\mathsf{H}} = \mathsf{poly}(\lambda)$ *and PPT adversary* $\mathcal{A}$ *that makes at most* Q_{H} *random oracle queries with*

$$\Pr\left[\begin{array}{c} (\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{X}_k, \pi^{\mathsf{m}}_k)\}_{k\in[\mathsf{Q}_{\mathsf{S}}]} \xleftarrow{\$} \mathcal{A}^{\mathsf{H}_{\mathsf{m}}}(1^\lambda, \widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}), \end{array} : \forall k \in [\mathsf{Q}_{\mathsf{S}}], \mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\widetilde{\mathsf{crs}}, \mathsf{X}_k, \pi^{\mathsf{m}}_k) = \top \right] \geq \mu(\lambda),$$

we have,

$$\Pr\left[\begin{array}{c} (\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^\lambda), \\ \{(\mathsf{X}_k, \pi^{\mathsf{m}}_k)\}_{k\in[\mathsf{Q_S}]} \xleftarrow{\$} \mathcal{A}^{\mathsf{H_m}}(1^\lambda, \widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}), \end{array} \begin{array}{r} \forall k \in [\mathsf{Q_S}], \mathsf{Verify}^{\mathsf{H_m}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}_k, \pi^{\mathsf{m}}_k) = \top \\ : \wedge \exists i \in [\tau], |\Gamma_{1,i}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq 3 \\ \wedge\ |\Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq \frac{\mu}{2\mathsf{Q_H}} |C_{\mathsf{ham}}| \end{array}\right]$$

is at least $\mu(\lambda)/2 - \mathsf{negl}(\lambda)$

Proof Sketch. For simplicity, denote $\Gamma^{(k)}_{1,i} := \Gamma_{1,i}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$ and $\Gamma^{(k)}_2 := \Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$ for each $(k, i) \in [\mathsf{Q_S}] \times [\tau]$. We denote by $\mathsf{ValidProofs}$ the event that $\mathsf{Verify}^{\mathsf{H_m}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}_k, \pi^{\mathsf{m}}_k) = \top$ for all $k \in [\mathsf{Q_S}]$. Then, to lower bound the desired probability, it suffices to upper bound $\sum_{k\in[\mathsf{Q_S}]} \Pr[\mathsf{ValidProofs} \wedge \forall i \in [\tau], |\Gamma^{(k)}_{1,i} < 3|]$ and $\sum_{k\in[\mathsf{Q_S}]} \Pr[\mathsf{ValidProofs} \wedge |\Gamma^{(k)}_2| < \frac{\mu}{2\mathsf{Q_H}} \cdot |C_{\mathsf{ham}}|]$. To obtain the bound on the later, observe that if $\Gamma^{(k)}_2$ has size at most T, then even a computationally unbounded (classical) adversary can find an input that hashes to $\Gamma^{(k)}_2$ with probability at most T/C_{ham} for every RO query. We can tune the size of T to get the desired bound. The bound on the former requires more work since at a high level the adversary can cheat twice; once for the second flow challenge and once for the fourth flow challenge. We show that if it cheats with respect to the second (resp. fourth) flow challenge then even a computationally unbounded (classical) adversary cannot cheat in the fourth (resp. second) flow challenge. □

We note that the main differences of the proof in the classical ROM and QROM is the bound in the statement of Lemma 4.1 and how it is proven. Informally, the reason why the above proof fails is because a quantum adversary can query the random oracle on all the input space in super position. To this end, we rely on (roughly) the optimality of the Grover's search to bound the success probability of the adversary.

We next show a restricted notion of the standard *special soundness* for interactive protocols. Typically, an extractor for special soundness is provided multiple valid transcripts containing the same commitments and is asked to extract a witness from them. Below, we show that for our particular interactive protocol, the extractor only requires one valid transcript along with several challenges for which existence of a valid response is guaranteed. Put differently, rather than taking multiple valid transcripts as input, our extractor only requires one transcript and the challenges included in the remaining valid transcripts. As explained in the overview of **??**, the crux of the proof is that given a valid challenge, the extractor can extract parts of the response by using the trapdoor τ (i.e., NTRU decryption key).

Lemma 4.2. *Consider the following 7 valid transcripts for a statement* X*:*

- *For* $(\eta, b) \in [3] \times [2]$, $\mathsf{trans}^{(\eta,b)} := \big(a_1, \mathbf{c}^{(\eta)}_1 := (c^{(\eta)}_i)_{i\in[\tau]}, a^{(\eta)}_2, \mathbf{c}^{(\eta,b)}_2 := (\boldsymbol{\beta}^{(\eta,b)} := (\beta^{(\eta,b)}_{i,j})_{(i,j)\in[\tau]\times[\tau']}, \beta'), \mathsf{resp}^{(\eta,b)}\big)$,
- $\widehat{\mathsf{trans}}^{(1,0)} := (a_1, \mathbf{c}^{(1)}_1, a^{(1)}_2, \widehat{\mathbf{c}}^{(1,b)}_2 := (\boldsymbol{\beta}^{(1,0)}, \widehat{\beta}'), \widehat{\mathsf{resp}}^{(1,0)})$,

such that there exists $(i^*, j_1^*, j_2^*, j_3^*) \in [\tau] \times [\tau']^3$ *that* $(c_{i^*}^{(1)}, c_{i^*}^{(2)}, c_{i^*}^{(3)})$ *are pairwise distinct,* $(\beta_{i^*,j_1^*}^{(1,0)}, \beta_{i^*,j_1^*}^{(1,1)})$, $(\beta_{i^*,j_2^*}^{(2,0)}, \beta_{i^*,j_2^*}^{(2,1)})$, *and* $(\beta_{i^*,j_3^*}^{(3,0)}, \beta_{i^*,j_3^*}^{(3,1)})$ *are each pairwise distinct, and* $\beta' \neq \widehat{\beta}'$.

Then, there exists a deterministic PT special sound extractor $\mathsf{Extract}_{\mathsf{ss}}$ *such that given a trapdoor* τ *to* $\widetilde{\mathsf{crs}}_{\mathsf{NIZK}}^{\mathsf{m}}$, *any statement* X *and* $\left(\mathsf{trans}^{(1,0)}, \left(\beta_{i^*,j_\eta^*}^{(\eta,0)}, \beta_{i^*,j_\eta^*}^{(\eta,1)}\right)_{\eta\in[3]}, \left(\beta', \widehat{\beta}'\right)\right)$ *included in any of the 7 valid transcripts of the above form,* $\mathsf{Extract}_{\mathsf{ss}}$ *outputs a witness* W *such that* $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}_{\mathsf{gap}}^{\mathsf{m}}$ *or a solution to the* $\mathsf{MSIS}_{d,1,k_4,16B_{\mathbf{Z}},q'}$ *problem with respect to* $\mathbf{a}_0 \in R_{q'}^{k_4}$ *included in* $\widetilde{\mathsf{crs}}_{\mathsf{NIZK}}^{\mathsf{m}}$ *or a solution to the* $\mathsf{MSIS}_{d,1,k_3,2(B_{\mathbf{Z}'}+B_c\delta^{\mathsf{gap}}),q'}$ *problem with respect to* $\mathbf{b}_0 \in R_{q'}^{k_3}$ *included in* $\mathsf{crs}_{\mathsf{com}}$.

Proof Sketch. The proof consists of three parts: in Part (A), we extract a witness that proves the linear relation (i.e., $\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\end{bmatrix}\mathbf{R}' + \begin{bmatrix}0\\ h\mathbf{g}\end{bmatrix}$); in Part (B), if the extracted witness from Part (A) is not in $\mathcal{R}_{\mathsf{gap}}^{\mathsf{m}}$, then we further extract a different witness that proves the exact relation for $\mathbf{t}_1$ (i.e., $\mathbf{t}_1 = \mathbf{b}_0\mathbf{R}''$); in Part (C), we show that given two different openings to $\mathbf{t}_1$, we can extract a solution to an MSIS problem. Looking ahead, if $\mathsf{Extract}_{\mathsf{ss}}$ does not succeed in outputting a valid witness for $\mathcal{R}_{\mathsf{gap}}^{\mathsf{m}}$ in Part (A), then it will only output a solution to the MSIS solution in the following Parts (B) and (C). This subtle observation will be used to optimize the proof size of our multi-proof extractable NIZK in the classical ROM.

Part (A). First observe that from $\mathsf{trans}^{(1,0)}$, we have

$$\bar{\mathbf{V}}' + \beta' \cdot \mathbf{V}' = H\mathbf{F}_1^{(1,0)'} + p\mathbf{F}_2^{(1,0)'} + \mathbf{Z}^{(1,0)'} \text{ (over } R_Q).$$

Notice the right hand side is a valid NTRU ciphertext. Namely, by using the trapdoor $\tau = (f, v)$ such that $H = p \cdot v \cdot f^{-1}$ (i.e., secret key for the NTRU encryption scheme), $\mathsf{Extract}_{\mathsf{ss}}$ can decrypt $\bar{\mathbf{V}}' + \beta' \cdot \mathbf{V}'$ to recover the "message" $\mathbf{Z}^{(1,0)'}$. Formally, $\mathbf{Z}^{(1,0)'} = f^{-1} \cdot (f \cdot (\bar{\mathbf{V}}' + \beta' \cdot \mathbf{V}') \mod Q) \mod p$. Moreover, by setting the parameters appropriately, the NTRU encyption scheme will have no decryption error. Thus, if $\bar{\mathbf{V}}' + \beta' \cdot \mathbf{V}'$ is guaranteed to be in the above form, then the possible $\mathbf{Z}^{(1,0)'}$ that can be included in $\mathsf{resp}^{(1,0)}$ is unique. In other words, there can not exist a distinct $\hat{\mathbf{Z}}^{(1,0)'}$ in $\mathsf{resp}^{(1,0)}$ such that verification still holds. The same argument holds for the $\zeta^{(1,0)}$ component since we have $\overline{v}' + \beta' \cdot v' = Hf_1^{(1,0)'} + pf_2^{(1,0)'} + \zeta^{(1,0)}$.

With this observation in mind, given $\mathsf{trans}^{(1,0)}$ and $\widehat{\beta}'$, $\mathsf{Extract}_{\mathsf{ss}}$ first performs NTRU decryption as follows, which is guaranteed to succeed by assumption:

$$\widehat{\mathbf{Z}}^{(1,0)'} := f^{-1} \cdot (f \cdot (\bar{\mathbf{V}}' + \widehat{\beta}' \cdot \mathbf{V}') \mod Q) \mod p,$$
$$\widehat{\zeta}^{(1,0)} := f^{-1} \cdot (f \cdot (\overline{v}' + \widehat{\beta}' \cdot v') \mod Q) \mod p.$$

As argued above, this $\widehat{\mathbf{Z}}^{(1,0)'}$ and $\widehat{\zeta}^{(1,0)}$ are guaranteed to be included in $\widehat{\mathsf{trans}}^{(1,0)}$, where note that $\widehat{\mathsf{trans}}^{(1,0)}$ is not provided to $\mathsf{Extract}_{\mathsf{ss}}$ as input. Since $\mathsf{trans}^{(1,0)}$

and $\widehat{\mathsf{trans}}^{(1,0)}$ are valid and share the same first flow a_1, they also satisfy the same verification equations regarding $\mathbf{w}'_1$ and $\mathbf{w}'_2$ (see Fig. 2). $\mathsf{Extract}_{\mathsf{ss}}$ subtracts these equations to remove $\mathbf{w}'_1$ and $\mathbf{w}'_2$, and obtains the following:

$$(\beta' - \widehat{\beta}') \cdot \mathbf{t}_1 = \mathbf{b}_0\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big) \text{ (over } R_{q'}),$$
$$(\beta' - \widehat{\beta}') \cdot \mathbf{t}_2\Delta = \mathbf{b}_1\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big)\Delta + [\zeta^{(1,0)} - \widehat{\zeta}^{(1,0)} \mid 0 \mid \cdots \mid 0] \text{ (over } R_q).$$

By multiplying Δ^{-1} from both sides in the later equation, $\mathsf{Extract}_{\mathsf{ss}}$ obtains

$$(\beta' - \widehat{\beta}') \cdot \mathbf{t}_2 = \mathbf{b}_1\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big) + (\zeta^{(1,0)} - \widehat{\zeta}^{(1,0)}) \cdot \mathbf{g}.$$

Due to our parameter selection, $(\beta' - \widehat{\beta}')$ is small and is guaranteed to be invertible over R_q. $\mathsf{Extract}_{\mathsf{ss}}$ then checks if $\mathbf{R}' := \big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big)/(\beta' - \widehat{\beta}')^{-1}$ consists of polynomials with $\{0,1,2\}$-coefficients. If so, $\mathsf{W} := ((\zeta^{(1,0)} - \widehat{\zeta}^{(1,0)}), (\beta' - \widehat{\beta}'), \mathbf{R}')$ is a valid witness for $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ and thus $\mathsf{Extract}_{\mathsf{ss}}$ outputs W.

We highlight again that if $\mathsf{Extract}_{\mathsf{ss}}$ does not succeed in outputting a valid witness for $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ in Part (A), then it can only output a solution to the MSIS problem in Parts (B) and (C). □

We are now ready to finish the proof of Theorem 4.1. The goal of Multi-Extract is to collect the necessary inputs to invoke $\mathsf{Extract}_{\mathsf{ss}}$ defined in Lemma 4.2. Let us informally explain in a bit more detail.

Given a valid proof π^{m}, Multi-Extract first goes over the challenges in C_{ham} to find another β'_t for which there exists a valid response. Concretely, it decrypts $(\overline{v}' + \beta'_t \cdot v')$ and $(\overline{\mathbf{V}}' + \beta'_t + \mathbf{V}')$ and searches for a pair $(\zeta_t, \mathbf{Z}'_t)$ that satisfies $\|\zeta_t\|_2 < B \wedge \|\mathbf{Z}'_t\|_2 < B_{\mathbf{Z}'} \wedge \mathbf{w}'_1 = \mathbf{b}_0\mathbf{Z}'_t - \beta'_t \cdot \mathbf{t}_1 \wedge \mathbf{w}'_2 = \mathbf{b}_1\mathbf{Z}'_t\Delta + [\zeta_t|0|\cdots|0] - \beta'_t \cdot \mathbf{t}_2\Delta$. If this is satisfied, $\mathsf{resp}_t = \big((\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}'_t, \zeta_t, f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\big)$ is guaranteed to be another valid response where the fourth flow challenge is $\mathbf{c}_{2,t} = (\boldsymbol{\beta}, \beta'_t)$. Note that this corresponds to $\widehat{\mathsf{resp}}^{(1,0)}$ and $\widehat{\beta}'$ in Lemma 4.2.

Multi-Extract then goes over *all* the challenges in C_X, which it can do since $|C_X| = 2d = \mathsf{poly}(\lambda)$. Concretely, for all $\beta \in C_X$, it decrypts $(\bar{\mathbf{V}}_{i',j'} + \beta \cdot \mathbf{V}_{i'})$ for all $(i', j') \in [\tau] \times [\tau']$, and checks if it correctly decrypts to some "message" $\mathbf{Z}_{\beta,i',j'}$ such that $\|\mathbf{Z}_{\beta,i',j'}\|_2 < B_{\mathbf{Z}}$. Note that unlike for the above set of challenges in C_{ham}, this check itself does not guarantee that there exists a valid transcript for challenge $\beta \in C_X$. This is because the fact that a valid $\mathbf{Z}_{\beta,i',j'}$ exists does not imply that there exists an associated valid third flow a_2. However, the main observation is that if a valid transcript for challenge $\beta \in C_X$ exists, then $(\bar{\mathbf{V}}_{i',j'} + \beta \cdot \mathbf{V}_{i'})$ must decrypt to $\mathbf{Z}_{\beta,i',j'}$ such that $\|\mathbf{Z}_{\beta,i',j'}\|_2 < B_{\mathbf{Z}}$.

Finally, Multi-Extract is ready to run $\mathsf{Extract}_{\mathsf{ss}}$. It runs through all three pairs of distinct challenges $\big(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\big)_{\eta\in[3]}$ it collected while going over C_X and executes $\mathsf{Extract}_{\mathsf{ss}}\big(\tau, \mathsf{X}, \big(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\big)_{\eta\in[3]}, (\beta, \widehat{\beta}')\big)$. We show via Lemmata 4.1 and 4.2 that with non-negligible probability, one of the set of inputs to $\mathsf{Extract}_{\mathsf{ss}}$ must be in the specified form detailed in Lemma 4.2. Moreover, $\mathsf{Extract}_{\mathsf{ss}}$ is only invoked a polynomially number of times. Thus, assuming the MSIS problem is difficult, Multi-Extract extracts a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ in polynomial time. □

4.3 Putting Everything Together

par.	q	q'	p	Q	τ	τ'	κ	d	k_1	k_2	k_3	k_4	B_c	σ	$\gamma_{\mathsf{DSMR}}, \gamma_{\mathbf{D}}, \gamma_{\mathbf{D}'}, \gamma_{\mathbf{E}}$
value	$\sim 2^{60}$	$\sim 2^{24}$	$\sim 2^{32}$	$\sim 2^{66}$	6	2	2	2048	3	5	4	19	36	2^{26}	1

Table 2: Concrete parameters for our scheme.

Roughly, we consider all the constraints that need to be satisfied by the correctness and security of our blind signature and use the LWE-Estimator from [6] so that every MLWE, MSIS, and DSMR assumptions give at least 128 bits of security. We employ the technique of Bai-Galbraith [11] to reduce the dimension of the signature by 2. We also consider that Gaussians can be encoded in $\log(2\sigma)$ bits by using the encoding of e.g. [45]. The size of the resulting signature is 102.6 KB and we get a first flow message of size 34 MB. However, as explained in the technical overview, we can reduce the first flow message in the classical ROM by removing the Katsumata transform [34] applied to the exact proof of Bootle et al. [16]. With this optimization, the first flow message is greatly reduced to 851 KB.

Possible optimizations. We also mention several possible optimizations. We can first consider using matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1$ instead of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1$ and lowering the degree d to e.g. 512. This can lower both the signature and first flow message size. This way we would have better granularity when modifying parameters, however we would need a module-NTRU trapdoor on the matrix $\mathbf{A}_1$ which is not constructed in [25] and seems nontrivial to obtain. Another solution would be to additionally prove the sparseness of $\mathbf{R}$ in the multi-proof extractable NIZK, which allows to lower the signature size since we will be able to extract $\mathbf{R}$ with better quality. This is possible by proving statements about the hamming weight of $\mathbf{R}$ but it would make the protocol much more complicated and the size of the first flow message may increase. Using either of these improvements we could lower the signature size to around 50 KB.

Another possible avenue for improvement would be reducing the size of the first flow by considering a better exact zero-knowledge proof. In all likelihood using the same proof as [28] would give the same improvement and bring the size of the first flow down to around 110 KB. However using this zero-knowledge proof is not completely straightforward as extraction is more complicated and the arguments used in Lemma 4.2 might not apply any more, especially when considering extraction in the QROM.

We leave further optimized instantiation of our generic construction as an interesting future work.

Acknowledgements. Shuichi Katsumata was partially supported by JSPS KAKENHI Grant Number 22K17892, Japan and JST AIP Acceleration Research JPMJCR22U5, Japan.

References

1. Vpn by google one, explained. https://one.google.com/about/vpn/howitworks.
2. M. Abe and T. Okamoto. Provably secure partially blind signatures. *CRYPTO 2000*, pp. 271–286.
3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. *EUROCRYPT 2010*, pp. 553–572.
4. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. *CRYPTO 2010*, pp. 98–115.
5. S. Agrawal, E. Kirshanova, D. Stehle, and A. Yadav. Can round-optimal lattice-based blind signatures be practical? *Cryptology ePrint Archive*.
6. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203.
7. N. A. Alkadri, P. Harasser, and C. Janson. Blindor: An efficient lattice-based blind signature scheme from or-proofs. In *CANS*, pp. 95–115. Springer.
8. N. Alkeilani Alkadri, R. El Bansarkhani, and J. Buchmann. BLAZE: Practical lattice-based blind signatures for privacy-preserving applications. *FC 2020*, pp. 484–502.
9. N. Alkeilani Alkadri, R. El Bansarkhani, and J. Buchmann. On lattice-based interactive protocols: An approach with less or no aborts. *ACISP 20*, pp. 41–61.
10. T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. *CRYPTO 2020, Part II*, pp. 470–499.
11. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. *CT-RSA 2014*, pp. 28–47.
12. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. *SCN 18*, pp. 368–385.
13. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. *ACM CCS 2006*, pp. 390–399.
14. D. Bernhard, M. Fischlin, and B. Warinschi. Adaptive proofs of knowledge in the random oracle model. *PKC 2015*, pp. 629–649.
15. W. Beullens, S. Dobson, S. Katsumata, Y.-F. Lai, and F. Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. *To Appear at EUROCRYPT*.
16. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. *CRYPTO 2019, Part I*, pp. 176–202.
17. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. *PKC 2010*, pp. 499–517.
18. S. Brands. Untraceable off-line cash in wallets with observers (extended abstract). *CRYPTO'93*, pp. 302–318.
19. J. Camenisch. Efficient and generalized group signatures. *EUROCRYPT'97*, pp. 465–479.
20. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *EUROCRYPT 2001*, pp. 93–118.
21. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *EUROCRYPT 2010*, pp. 523–552.
22. D. Chaum. Blind signatures for untraceable payments. *CRYPTO'82*, pp. 199–203.
23. D. Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. *EUROCRYPT'88*, pp. 177–182.

24. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. *CRYPTO'88*, pp. 319–327.
25. C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, and K. Xagawa. ModFalcon: Compact signatures based on module-NTRU lattices. *ASIACCS 20*, pp. 853–866.
26. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. *ACM CCS 2018*, pp. 574–591.
27. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268.
28. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. *ASIACRYPT 2020, Part II*, pp. 259–288.
29. M. F. Esgin, R. Steinfeld, D. Liu, and S. Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. *Cryptology ePrint Archive*.
30. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. *CRYPTO 2006*, pp. 60–77.
31. A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT*, pp. 244–251. Springer.
32. S. Garg, V. Rao, A. Sahai, D. Schröder, and D. Unruh. Round optimal blind signatures. *CRYPTO 2011*, pp. 630–648.
33. E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-based blind signatures, revisited. *CRYPTO 2020, Part II*, pp. 500–529.
34. S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. *CRYPTO 2021, Part II*, pp. 580–610, Virtual Event, 2021.
35. H. Q. Le, W. Susilo, T. X. Khuc, M. K. Bui, and D. H. Duong. A blind signature from module latices. In *Dependable and Secure Computing (DSC)*, pp. 1–8. IEEE.
36. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. *ASIACRYPT 2009*, pp. 598–616.
37. V. Lyubashevsky. Lattice signatures without trapdoors. *EUROCRYPT 2012*, pp. 738–755.
38. V. Lyubashevsky, N. K. Nguyen, and M. Plancon. Efficient lattice-based blind signatures via gaussian one-time signatures. *To Appear at PKC*.
39. V. Lyubashevsky, N. K. Nguyen, and M. Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *To Appear at Crypto*.
40. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. *PKC 2021, Part I*, pp. 215–241.
41. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. *EUROCRYPT 2012*, pp. 700–718.
42. T. Okamoto and K. Ohta. Universal electronic cash. *CRYPTO'91*, pp. 324–337.
43. D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. Cryptology ePrint Archive, Report 2019/1452.
44. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396.
45. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. Technical report, 2018. Available at https://falcon-sign.info/.
46. M. Rückert. Lattice-based blind signatures. *ASIACRYPT 2010*, pp. 413–430.

47. C.-P. Schnorr. Security of blind discrete log signatures against interactive attacks. *ICICS 01*, pp. 1–12.
48. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *EUROCRYPT'98*, pp. 1–16.
49. X. Yi and K.-Y. Lam. A new blind ECDSA scheme for bitcoin transaction anonymity. *ASIACCS 19*, pp. 613–620.
50. M. Zhandry. How to construct quantum random functions. In *53rd FOCS*, pp. 679–687.