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

June 23, 2022

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.

Contents

1 Introduction

1.1 Background

Blind signatures, originally proposed by Chaum [Cha82], 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 [Cha82, CFN90, OO92], anonymous credentials [Bra94, CL01], e-voting [Cha88, FOO92], and so on, and more recently, it has found exciting applications in the context of adding privacy features to blockchains [YL19] and privacy-preserving authentication tokens [Goo22].

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 (see Section 1.4 for other related works). The first lattice-based blind signature was proposed by Rükert [Rüc10], who followed a design paradigm similar to the classical Schnorr or Okamoto-Schnorr blind signatures [Sch01, PS00]. 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 [PHBS19, LSK+19, AEB20a, AEB20b, AHJ21], where BLAZE+ by Alkadri et al. [AEB20b] currently stands as the most efficient proposal. However, recently, Hauck et al. [HKLN20] showed that all constructions following the blueprint of Rükert's blind signature contain the same bug in their security proof[1], 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. [HKLN20] 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. [AKSY21a]. Instead of following the three-move structure seen in Schnorr's blind signature [Sch01], Agrawal et al. builds on Fischlin [Fis06] and Garg et al. [GRS+11] 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 SIS assumption. Moreover, the signature size becomes as large as 1 MB[2,3], while the communication cost is lowered to a few KB. The other work is by Lyubashevsky et al. [LNP22a]. They propose a round-optimal blind signature based on a new approach using one-time signatures and OR-proofs. Unlike [AKSY21a], 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

[1] Alkadri et al. [AHJ21] 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.

[2] Agrawal et al. provide an informal estimate of 30 KB to 100 KB and states to use the NIZK by [ENS20, LNS21]. 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.

[3] After submission of this paper, Agrawal et al. updated their paper to use the NIZK by Lyubashevsky et al. [LNP22b] appearing at CRYPTO 2022. See Section 1.4 work for more detail.

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) 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 [AO00]. 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 [Cha82, CFN90, OO92] and e-voting [Cha88, FOO92].

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 [Fis06]. 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. [dLS18] 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 [Lyu09, Lyu12] to generate the signature.

One tool required by our generic construction is a *multi-proof straight-line extractable* NIZK [BDK+21],[4] 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 [PS00, BN06]. 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 [SG98, BFW15] and group signatures [BDK+21]. 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 [Kat21].

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 [ALS20, ENS20], then we could

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

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.

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 [Fis06]. 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

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

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. [AKSY21a] attempts to heuristically[5] instantiate Fischlin's generic construction based on Dilithium [DKL+18], 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. [dLS18] 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 [Cam97]. Each group user is assigned an identity $I \in [N]$, where $N = \mathsf{poly}(\lambda)$ 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).

[5] 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.

We briefly go over the group signature by del Pino et al. [dLS18]. They use Boyen's lattice-based signature [Boy10, ABB10b] 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 [MP12]. 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. [BDL+18], 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 $\Sigma = (\pi, \mathsf{com})$.[6]

Reversing the Order for Blind Signatures. The technique of del Pino et al. [dLS18] 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.[7] 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

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

[6] 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.

[7] A keen reader may notice that the message space (i.e. group size) $[N]$ has to be polynomial large for the security proof of [dLS18] to work. We later show how to support an exponentially large message space as required for blind signatures.

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 [ABB10a, CHKP10] 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. a valid BDLOP commitment) and

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

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 [ABB10a, CHKP10] 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 [PS00, BN06], 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 Section 1.2, we rely on a stronger type of *multi-proof straight-line extractable* NIZK [BDK+21]. 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 [ABB10a, CHKP10] 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 7, 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 [Zha12] 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 [ABB10a, CHKP10]. 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 [Kat21]. 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 [Lyu09, Lyu12] 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. [BLS19] 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. [ESLR22]. 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. [BLS19] has a larger proof size compared to the standard NIZK for MSIS/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 [ALS20, ENS20] 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. Details are provided in Section 4.4.

1.4 Related Work

Blind Signatures in the Standard Model. Blind signature have been the target of many theoretical works since they are a special case of a general two-party computation. Lindell [Lin08], Fischlin and Schröder [FS10], and Pass [Pas11] all show some impossibility results on blind signatures in less than three rounds in the standard model, i.e. without using a common reference string (CRS) or relying on the ROM. Garg et al. [GRS+11] constructed the first round-optimal blind signature in the standard model, where they circumvent the impossibility result by using complexity leveraging. Fuchsbauer et al. [FHS15] constructed an efficient round-optimal blind signature in the standard model relying on interactive assumptions. Katsumata et al. [KNYY21] constructed the first round-optimal blind signature in the standard model without using complexity leveraging. They circumvent the impossibility result by using a quantum reduction to break classical assumptions in the security proof.

Blind Signatures in the ROM/CRS from Classical Assumptions. Fischlin [Fis06] proposed a generic construction of a round-optimal blind signature in the CRS model. Schnorr [Sch01] constructed an efficient three-round blind signature based on the Schnorr signature [Sch90]. Although the Schnorr blind signature was considered to be secure against poly-logarithmically many signature, it was not until recently that provable security in the algebraic group model (AGM) and ROM was established [FPS20, KLX20]. Baldimtsi and Lysyanskay [BL13] showed that proving the Schnorr blind signature only in the ROM is impossible. Pointcheval and Stern [PS00] proved that the three-round Okamoto-Schnorr blind signature based on the DDH assumption is secure for poly-logarithmically many signatures. Abe and Okamoto [AO00] introduced the concept of *partial* blind signatures and constructed a three-round blind signature based on the DDH assumption that is secure for poly-logarithmically many signatures. It was recently shown by Benhamouda et al. [BLL+21] that there is a practical attack on [Sch01, PS00, AO00] when the number of signatures exceeds the amount supported by their respective security proofs. Abe [Abe01],[KLX20] constructed a three-round blind signature in the AGM that is secure for polynomially many signatures. Tessaro and Zhu [TZ22] recently constructed a blind signature with similar properties but with signature size one-half of the Abe blind signature.

Concurrent and Independent Work. In a recent series of work [BLS19, ALS20, LNS20, ENS20, LNP22b], increasingly tight and efficient exact lattice-based zero-knowledge proofs have been constructed. In this paper we do not use the latest of these improvements (we use [BLS19] and not the very recent [LNP22b]), first because the efficiency of our exact NIZK does not affect the final signature size (as it is only necessary when sending the first flow to the signer), and also because using more involved proofs of knowledge would make the security proof more complicated and the paper less readable, we thus leave this task to a future work.

While, as mentioned, the efficiency of our exact proof does not impact the signature size, its tightness does. In fact using [LNP22b] which proves tight bounds on the euclidean norm (rather than [BLS19] which proves bounds on the infinity norm) would help improve the parameters of our scheme, it could even be worthwhile to replace our second NIZK (which does not need to be exact for security) with this new proof, as having a less efficient but tighter proof would result in even better parameters and potentially smaller signatures. Agrawal et al. [AKSY21b] constructs a blind signature also based on the Fischlin blind signature relying on a new assumption called one-more-SIS. After the submission of this paper, [AKSY21b] updated their paper to include a parameter set achieving signature size $44KB$ using the new NIZK of [LNP22b]. The claimed security is 109 bits and while the paper provides some potential attack directions, the new one-more-SIS assumption warrants further cryptanalysis from the community.

2 Preliminaries

Notations. For sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathsf{Func}(\mathcal{X},\mathcal{Y})$ denotes the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We view vectors $\mathbf{a}$ in their row form. For two vectors $\mathbf{a}$ and $\mathbf{b}$, $[\mathbf{a}^\top \| \mathbf{b}^\top]$ denotes the vertical concatenation. We use PPT and QPT as shorthand for probabilistic polynomial time and quantum polynomial time, respectively.

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*

$$\Pr\left[\begin{array}{l}(\mathsf{vk},\mathsf{sk}) \xleftarrow{\$} \mathsf{BSGen}(1^\lambda)\\ (\rho_1, \mathsf{st}_{\mathcal{U}}) \xleftarrow{\$} \mathcal{U}_1(\mathsf{vk}, \mathsf{M})\\ \rho_2 \xleftarrow{\$} \mathcal{S}_2(\mathsf{sk}, \rho_1)\\ \Sigma \xleftarrow{\$} \mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2)\end{array} : \mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}, \Sigma) = \top\right] = 1 - \mathsf{negl}(\lambda).$$

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, we have*

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

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*

$$\mathsf{Adv}^{\mathsf{blind}}_{\Pi_{\mathsf{BS}}}(\mathcal{A}) := \left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right| = \mathsf{negl}(\lambda).$$

Remark 2.5 (Blind Signature in the (Q)ROM). In the (Q)ROM, we assume all algorithms used to define Π_{BS} and the adversary are provided oracle access to the random oracle. For instance, in the game of one-more unforgeability, we assume $\mathcal{A}$ and $\mathcal{S}_2$ to have access to the random oracle. All probabilities are also taken over the random choice of the random oracle.

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 chose to make the reliance on the (Q)ROM explicit for NIZKs unlike for other primitives considered in the paper such as blind signatures since the definition deviates slightly from those in the standard model. We also 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. Below, we define NIZKs with respect to quantum adversaries but we can recover the classical definition by restricting the adversaries to be classical.

Definition 2.6 (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)*[8] *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 $\mathcal{R}$.

We require an NIZK proof system to satisfy several properties. Below, we always assume probabilities are also taken over the random choices of the random oracle. We first consider correctness.

[8] 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 2.7 (Correctness). *An* NIZK *proof system* Π_{NIZK} *is correct if for all* $\lambda \in \mathbb{N}$, $\mathsf{crs} \in \{0,1\}^\ell$ *and* $(\mathsf{X},\mathsf{W}) \in R$, *the probability of* $\mathsf{Prove}^{\mathcal{O}}(\mathsf{crs},\mathsf{X},\mathsf{W})$ *outputting* $\bot$ *is at most* $\mathsf{negl}(\lambda)$, *and we have*

$$\Pr\left[\ \pi \xleftarrow{\$} \mathsf{Prove}^{\mathcal{O}}(\mathsf{crs},\mathsf{X},\mathsf{W}) : \mathsf{Verify}^{\mathcal{O}}(\mathsf{crs},\mathsf{X},\pi) = \top \ \middle|\ \pi \neq \bot\right] = 1.$$

We consider the standard notion of zero-knowledge, except that we assume that an adversary only obtains at most two proofs per statement. This is sufficient for blind signatures and simplifies our proof for zero-knowledge in the QROM (see Footnote 11 for more detail). Note that for (deterministic) Fiat-Shamir-based signature schemes in the QROM [KLS18, Kat21], it suffices to assume that the adversary can receive a single proof per statement.

Definition 2.8 (Zero-Knowledge). *An* NIZK *proof system* Π_{NIZK} *is classically (resp. quantumly)* zero-knowledge *if there exists a PPT zero-knowledge simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *consisting of two algorithms* Sim_0 *and* Sim_1 *with a shared state such that for any PPT (resp. QPT) adversary* $\mathcal{A}$, *we have*

$$\mathsf{Adv}^{\mathsf{ZK}}_{\Pi_{\mathsf{NIZK}}}(\mathcal{A}) := \left|\Pr\left[\mathcal{A}^{|\mathcal{O}\rangle,\mathsf{Prove}}(\mathsf{crs}) = 1\right] - \Pr\left[\mathcal{A}^{|\mathsf{Sim}_0\rangle,\mathcal{S}}(\mathsf{crs}) = 1\right]\right| = \mathsf{negl}(\lambda),$$

where Prove *and* $\mathcal{S}$ *are prove oracles that on input* (X,W) *return* $\bot$ *if* $(\mathsf{X},\mathsf{W}) \notin \mathcal{R}$ *and otherwise return* $\mathsf{Prove}^{\mathcal{O}}(\mathsf{crs},\mathsf{X},\mathsf{W})$ *or* $\mathsf{Sim}_1(\mathsf{crs},\mathsf{X})$, *respectively. The probability is also taken over the randomness of sampling* $\mathsf{crs} \xleftarrow{\$} \{0,1\}^\ell$. *Here, we assume* $\mathcal{A}$ *queries the same statement* X *to* Prove *or* $\mathcal{S}$ at most twice.

We define proof of knowledge which is a stronger property than soundness. Informally, we require the existence of an extractor algorithm $\mathsf{Extract}$ such that for any adversary outputting a valid statement and proof pair, $\mathsf{Extract}$ can extract a corresponding witness. We can consider several flavors for proof of knowledge. Below, we consider two types: *single-proof extractability* and *multi-proof (straight-line) extractability*. While the latter is a stronger property compared to the former, the former allows for more efficient constructions.

The following single-proof extractability definition is identical to the standard definition of (non-adaptive) proof of knowledge.

Definition 2.9 (Single-Proof Extractability). *An* NIZK *proof system* Π_{NIZK} *is classically (resp. quantumly)* single-proof extractable *if there exists a PPT (resp. QPT) extractor* $\mathsf{Single\text{-}Extract}$, *constants* c_1, c_2, e, *and a non-negligible polynomial* $p(\lambda)$ *such that for any* $\mathsf{crs} \in \{0,1\}^\ell$, *any* $\mathsf{X} \in \mathcal{L}_{\mathcal{R}}$, *any* $\mathsf{Q_H} = \mathsf{poly}(\lambda)$, *and PPT (resp. QPT) adversary* $\mathcal{A}$ *that makes at most* $\mathsf{Q_H}$ *random oracle queries with*

$$\Pr[\pi \xleftarrow{\$} \mathcal{A}^{|\mathcal{O}\rangle}(\mathsf{crs},\mathsf{X}) : \mathsf{Verify}^{\mathcal{O}}(\mathsf{crs},\mathsf{X},\pi) = \top] \geq \mu(\lambda),$$

we have,

$$\Pr\left[\ \mathsf{W} \xleftarrow{\$} \mathsf{Single\text{-}Extract}^{\mathcal{A}}(\mathsf{crs},\mathsf{X}) \ \ : \ \ (\mathsf{X},\mathsf{W}) \in \mathcal{R}_{\mathsf{gap}}\ \right] \geq \frac{1}{p(\lambda)\cdot \mathsf{Q_H^e}} \cdot \mu(\lambda)^{c_1} - \mathsf{negl}(\lambda),$$

where the runtime of $\mathsf{Single\text{-}Extract}$ *is upper bounded by* $c_2 \cdot \mathsf{Time}(\mathcal{A})$ *and we assume one oracle access to* $\mathcal{A}$ *takes* $\mathsf{Time}(\mathcal{A})$.

For instance, in the classical setting, if we compile a sigma protocol with the Fiat-Shamir transform, then we have $(c_1, c_2, e) = (2, 2, 1)$ and $p(\lambda) = 1$ via the forking lemma [PS00, BN06]. In the quantum setting, [DFMS19, LZ19] showed that $(c_1, c_2, e) = (3, 2, 6)$ for some non-negligible $p(\lambda)$ if the sigma protocol is additionally *collapsing* (see Appendix B for more details).

We additionally rely on a stronger type of extractability where we can directly extract from multiple statement and proof pairs output by the adversary. Unlike the above definition, the adversary is further allowed to chose the statement adaptively. To perform such strong form of extraction, the common random string $\widetilde{\mathsf{crs}}$ is simulated and the extractor is provided with a special trapdoor corresponding to $\widetilde{\mathsf{crs}}$.

Definition 2.10 (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* $\mathcal{A}$*, we have*

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

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*

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

we have,

$$\Pr\left[\begin{array}{c} (\widetilde{\mathsf{crs}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^{\lambda}), \{(\mathsf{X}_i, \pi_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}]} \xleftarrow{\$} \mathcal{A}^{|\mathcal{O}\rangle}(\widetilde{\mathsf{crs}}), \\ \{\mathsf{W}_i \xleftarrow{\$} \mathsf{Multi\text{-}Extract}(1^{\lambda}, \mathsf{Q}_{\mathsf{H}}, \mathsf{Q}_{\mathsf{S}}, 1/\mu, \tau, \mathsf{X}_i, \pi_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}]} \end{array} : \begin{array}{l} \forall i \in [\mathsf{Q}_{\mathsf{S}}], (\mathsf{X}_i, \mathsf{W}_i) \in \mathcal{R}_{\mathsf{gap}} \\ \wedge\ \mathsf{Verify}^{\mathcal{O}}(\widetilde{\mathsf{crs}}, \mathsf{X}_i, \pi_i) = \top \end{array}\right] \geq \frac{1}{2}\cdot\mu(\lambda) - \mathsf{negl}(\lambda).$$

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 $(c, e_1, e_2) = (1, 2, 1)$.

Remark 2.11 (Regarding Common Random String). We only require a common random string crs for multi-proof extraction, and thus omit crs from the syntax for simplicity when only requiring single-proof extraction. Looking ahead, in the context of blind signatures, the crs is simply generated as an output of the random oracle since it is a common *random* string.

2.3 Lattices

Rings and Gaussian Measures. For a power of 2 integer d and a prime q, let R_q denote the polynomial ring $\mathbb{Z}_q[X]/(X^d+1)$. Throughout this paper we view ring elements $a = \sum_{i=0}^{d-1}\alpha_i X^i \in \mathbb{Z}[X]/(X^d+1)$ as row vectors $(\alpha_0, \cdots, \alpha_{d-1}) \in \mathbb{Z}^d$ interchangeably. 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 $[a,b]$. For a positive real σ, let $D_{\mathbb{Z}^d,\sigma}$ denote the discrete Gaussian distribution over $\mathbb{Z}^d$. For any $\mathbf{x} \in \mathbb{Z}^d$:

$$D_{\mathbb{Z}^d,\sigma}(\mathbf{x}) = \frac{\exp(\|\mathbf{x}\|_2^2/(2\sigma^2))}{\sum_{\mathbf{y}\in\mathbb{Z}^d}\exp(\|\mathbf{y}\|_2^2/(2\sigma^2))}$$

To simplify notations, we occasionally use $a \xleftarrow{\$} D_\sigma$ to mean that the coefficient vector of $a \in R_q$ is sampled from $D_{\mathbb{Z}^d,\sigma}$. The definitions naturally extends to vectors $\mathbf{a} \in R^k$ by viewing $\mathbf{a}$ as a vector in $\mathbb{Z}_q^{kd}$. Finally, for a matrix $\mathbf{R} \in \mathbb{Z}^{n\times m}$, we denote by $s_1(\mathbf{R})$ its spectral norm. We extend the notion to matrices over R by considering the coefficient embedding into $\mathbb{Z}$.

The following is the rejection sampling lemma by [Lyu12, Lemmas 4.3, 4.6].

Lemma 2.12 (Rejection Sampling). *Let* $V \subset \mathbb{Z}^m$ *in which all elements have* ℓ_2*-norm less than* T*,* h *be a probability distribution over* V*,* ϕ *a positive real,* err *a positive real smaller than 1, and set* $\sigma = \phi\cdot T$*. Now sample* $\mathbf{e} \leftarrow h$ *and* $\mathbf{r} \leftarrow D_{\mathbb{Z}^m,\sigma}$*, set* $\mathbf{z} = \mathbf{e}+\mathbf{r}$*, and run* $b \leftarrow \mathsf{Rej}(\mathbf{z}, \mathbf{e}, \phi, T, \mathsf{err})$ *in Fig. 1. Then, the probability that* $b = \top$ *is at least* $(1-\mathsf{err})/\mu(\phi, \mathsf{err})$ *for* $\mu(\phi, \mathsf{err}) = \exp\left(\sqrt{\frac{-2\log\mathsf{err}}{\log e}}\cdot\frac{1}{\phi} + \frac{1}{2\phi^2}\right)$ *and the distribution of* $(\mathbf{e}, \mathbf{z})$ *conditioned on* $b = \top$ *is within statistical distance of* $\mathsf{err}/\mu(\phi, \mathsf{err})$ *of the product distribution* $h \times D_{\mathbb{Z}^m,\sigma}$*.*

$\underline{\mathsf{Rej}(\mathbf{z}, \mathbf{e}, \phi, T, \mathsf{err})}$

1: $u \xleftarrow{\$} [0, 1)$
2: **if** $u > \frac{1}{\mu(\phi, \mathsf{err})} \cdot \exp\left(\frac{-2\langle \mathbf{z}, \mathbf{e}\rangle + \|\mathbf{e}\|_2^2}{2\sigma^2}\right)$ **then return** $\bot$
3: **else return** $\top$

Figure 1: Rejection sampling.

As a concrete example that is often used, by setting $\phi = 11$ and $\mathsf{err} = 2^{-100}$ we get $\mu(\phi, \mathsf{err}) \approx 3$. We can also set for example $\phi = 14$ and $\mathsf{err} = 2^{-256}$ to obtain $\mu(\phi, \mathsf{err}) \approx 4$ if we want better statistical bounds.

The following establishes useful lemmas to bound the norm of an element sampled from some discrete Gaussian distribution [MR04, Lyu12, ABB10a].

Lemma 2.13. *For any real $t > 0$ and $t' > 1$, we have*

$$\Pr[\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^n, \sigma} : \|\mathbf{x}\|_\infty > t\sigma] < 2n \cdot 2^{-\frac{\log e}{2} \cdot t^2},$$
$$\Pr[\mathbf{x} \xleftarrow{\$} D_{\mathbb{Z}^n, \sigma} : \|\mathbf{x}\|_2 > t\sigma\sqrt{n}] < 2^{n \cdot \left(\frac{\log e}{2}(1-t^2) + \log t\right)}.$$

Lemma 2.14. *Let k, q be positive integers larger than 2, $\mathbf{a} \in R_q^k$, $u \in R_q$, $\mathbf{T_a} \in R^{k \times k}$ be an arbitrary basis for $\Lambda^\bot(\mathbf{a})$, and $\sigma > \|\mathbf{T_a}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log kd})$. Then, if we sample a vector $\mathbf{e} \leftarrow D_{\Lambda_u^\bot(\mathbf{a}), \sigma}$, we have $\Pr[\|\mathbf{e}\|_2 > \sqrt{kd}\sigma] < \mathsf{negl}(d)$.*

The following states that with overwhelming probability, the MSIS problem has several solutions. The proof is a simple adaptation of [Lyu12, Lemma 5.2.] to the structured lattice setting.

Lemma 2.15. *Let d, k, q, Δ be positive integers and $R_q = \mathbb{Z}[X]/(X^d + 1)$. For any $\mathbf{a} \in R_q^k$ and $\mathbf{s} \xleftarrow{\$} [-\Delta, \Delta]_{\mathsf{coeff}}^k$, the probability that there does not exist $\mathbf{s}' \in [-\Delta, \Delta]_{\mathsf{coeff}}^k$ such that $\mathbf{s}' \neq \mathbf{s}$ and $\mathbf{a}\mathbf{s}^\top = \mathbf{a}\mathbf{s}'^\top$ is at most $q^d/(2 \cdot \Delta + 1)^{kd}$.*

Hardness Assumptions. We define several hardness assumptions used in this paper. We first define the module short integer solutions (MSIS) and module learning with errors (MLWE) assumption. Below, we assume the assumptions are difficult for QPT adversaries by default.

Definition 2.16 (MSIS). *For integers $d = d(\lambda), n = n(\lambda), k = k(d, n), q = q(d, n) > 2$, $B = B(d, n)$, and an algorithm $\mathcal{A}$, the advantage of the* module short integer solutions $\mathsf{MSIS}_{d,n,k,B,q}$ *problem of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{MSIS}_{d,n,k,B,q}}(\mathcal{A}) = \Pr[\mathcal{A}(\mathbf{A}) \to \mathbf{e} : 0 < \|\mathbf{e}\|_2 \leq B \wedge \mathbf{A}\mathbf{e}^\top = \mathbf{0} \mod q]$$

where $\mathbf{A} \xleftarrow{\$} R_q^{n \times k}$. We say the $\mathsf{MSIS}_{d,n,k,B,q}$ assumption holds if the above advantage is negligible for all QPT $\mathcal{A}$.

Definition 2.17 (MLWE). *For integers $d = d(\lambda), n = n(\lambda), k = k(d, n), q = q(d, n) > 2$, an error distribution $\chi = \chi(d, n)$ over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, and an algorithm $\mathcal{A}$, the advantage of the* module learning with errors $\mathsf{MLWE}_{d,n,k,\chi,q}$ *problem of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{MLWE}_{d,n,k,\chi,q}}(\mathcal{A}) = \left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s}^\top + \mathbf{e}^\top) \to 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{b}^\top) \to 1]\right|,$$

where $\mathbf{A} \xleftarrow{\$} R_q^{n \times k}$, $\mathbf{s} \xleftarrow{\$} \chi^k$, $\mathbf{e} \xleftarrow{\$} \chi^n$, and $\mathbf{b} \xleftarrow{\$} R_q^n$. We say the $\mathsf{MLWE}_{d,n,k,\chi,q}$ assumption holds if the above advantage is negligible for all QPT $\mathcal{A}$.

Finally, we define the decisional small *matrix* ratio (DSMR) assumption [CPS+20, Kat21] that generalizes the decisional small *polynomial* ratio (DSPR) assumption used by [HPS98, LTV12, SXY18]. The latter underlies the hardness of the NTRU encryption scheme.

Definition 2.18 (DSMR). *For integers $d = d(\lambda), k = k(d), p = p(d), q = q(d) > 2$ such that p and q are coprime, an error distribution $\chi = \chi(d)$ over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, and an algorithm $\mathcal{A}$, the advantage of the* decisional small matrix ratio $\mathsf{DSMR}_{d,k,\chi,q,p}$ *problem of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{DSMR}_{d,k,\chi,q,p}}(\mathcal{A}) = \left|\Pr[\mathcal{A}(p \cdot \mathbf{v} \cdot \mathbf{F}^{-1}) \to 1] - \Pr[\mathcal{A}(\mathbf{h}) \to 1]\right|,$$

where $(\mathbf{v}, \mathbf{F}) \leftarrow \chi^k \times \chi^{k \times k}$ conditioned on $\mathbf{F}$ being invertible over mod q *and* mod p, *$\mathbf{h} \leftarrow R_q^k$. We say the $\mathsf{DSMR}_{d,k,\chi,q,p}$ assumption holds if the above advantage is negligible for all QPT $\mathcal{A}$.*

Sampling Algorithms. Chuengsatiansup et al. [CPS+20] shows how to generate a lattice trapdoor based on the DSMR assumption. Although we can generate a lattice trapdoor without any computational assumptions or using only the MLWE assumption, e.g. [GPV08, MP12], relying on the DSMR assumption results in better parameters.

Lemma 2.19 (Trapdoor Generation). *Let $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ with d a power of 2, q a prime, and $k \geq 2$ a positive integer. Let $\chi := D_{\mathbb{Z},\sigma}$ for $\sigma \lesssim O(q^{1/k})$ for which the $\mathsf{DSMR}_{d,k-1,\chi,q,1}$ assumption holds. Then, there exists a randomized algorithm $\mathsf{TrapGen}(1^{kd}, q)$ that outputs a vector $\mathbf{a} := [1 \mid \mathbf{a}'] \in R_q^k$ and a full-rank matrix $\mathbf{T_a} \in R^{k \times k}$, where $\mathbf{T_a}$ is a basis for $\Lambda^\perp(\mathbf{a})$. Moreover, $\|\mathbf{T_a}\|_{\mathsf{GS}} = O(q^{1/k})$ and $\mathbf{a}' \in R_q^{k-1}$ is indistinguishable from random based on the $\mathsf{DSMR}_{d,k-1,\chi,q,1}$ assumption.*

Using a lattice trapdoor, we can perform the following types of discrete Gaussian sampling [ABB10a, CHKP10, CHKP12, MP12]. We modify SampleRight from [MP12] so that the so-called "MP-trapdoor" $\mathbf{R}^*$ can be large in a controlled manner. We believe this may have other applications and provide a proof sketch in Appendix A.1.

Lemma 2.20 (Trapdoor Sampling). *Let $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ with d a power of 2, q a prime, and k, k', k_1, k_2 positive integers such that $k, k' \geq 2$ and $k_1 + k_2 = k'$. Then, we have the following.*

- $\mathsf{SampleLeft}(\mathbf{a}, \mathbf{b}, u, \mathbf{T_a}, \sigma) \to \mathbf{e}$: *There exists a randomized algorithm that, given vectors $\mathbf{a} \in R_q^k$ and $\mathbf{b} \in R_q^{k'}$ with $k' = k_1 + k_2$, a ring element $u \in R_q$, a basis $\mathbf{T_a} \in R^{k \times k}$ for $\Lambda^\perp(\mathbf{a})$, and a Gaussian parameter $\sigma > \|\mathbf{T_a}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log kd})$, outputs a vector $\mathbf{e} \in R^{k+k'}$ sampled from a distribution which is $\mathsf{negl}(d)$-close to $D_{\Lambda_u^\perp([\mathbf{a}|\mathbf{b}]),\sigma}$.*
- $\mathsf{SampleRight}(\mathbf{a}, \mathbf{g}, (\mathbf{R}, c, \mathbf{R}'), t, u, \mathbf{T_g}, \sigma) \to \mathbf{e}$: *There exists a randomized algorithm that, given vectors $\mathbf{a} \in R_q^{k'}$, $\mathbf{g} \in R_q^k$, matrices $\mathbf{R} \in R^{k_1 \times k}$ and $\mathbf{R}' \in R^{k_2 \times k}$, invertible elements $c, t \in R_q$, a basis $\mathbf{T_g}$ for $\Lambda^\perp(\mathbf{g})$, and a Gaussian parameter $\sigma > s_1(c\mathbf{R}^*) \cdot \|\mathbf{T_g}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log kd})$, where $\mathbf{R}^* = \begin{bmatrix} \mathbf{R} \\ \frac{1}{c}\mathbf{R}' \end{bmatrix} \in R^{k' \times k}$, outputs a vector $\mathbf{e} \in R^{k'+k}$ sampled from a distribution which is $\mathsf{negl}(d)$-close to $D_{\Lambda_u^\perp([\mathbf{a}|\mathbf{aR}^*+t\cdot\mathbf{g}]),\sigma}$.*

In the above SampleRight algorithm, it is conventional to set $\mathbf{g}$ as the so-called "gadget matrix" [MP12]. For any integer $b \geq 2$, $\mathbf{g} := [1 \mid b \mid \cdots \mid b^{k-1}] \in R_q^k$, where $k = \lceil \log_b(q) \rceil$. The size of $\mathbf{g}$ is parameterized by b. Moreover, there exists a public known trapdoor $\mathbf{T_g} \in R^{k \times k}$ such that $\|\mathbf{T_g}\|_{\mathsf{GS}} \leq \sqrt{b^2 + 1}$.

2.4 Commitments

We provide a minimal definition for a commitment scheme in the common random string model. Below, we do not define the blinding property as it will be implicitly handled by the *trapdoor-sampling-compatibility* notion that we define in Section 3.1.

Definition 2.21 (Commitment Scheme). *A commitment scheme Π_{Com} with message space $\mathcal{M}$, randomness space $\mathcal{R}$ and common random string* crs *with length $\ell(\lambda)$ consists of the algorithm* Com *defined as follows:*

$\mathsf{Com}(\mathsf{crs}, \mathsf{M}; \mathsf{rand}) \to \mathsf{com}$: *The commitment algorithm takes as input the common random string* crs*, a message* $\mathsf{M} \in \mathcal{M}$*, and randomness* $\mathsf{rand} \in \mathcal{R}$*, and outputs a commitment* com*. We may omit* rand *when we do not require the randomness to be explicit.*

We require the commitment scheme to satisfy hiding.

Definition 2.22 (Hiding). *A commitment scheme with message space* $\mathcal{M}$ *is classically (resp. quantumly)* hiding *if for any PPT (resp. QPT) algorithm* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathsf{hide}}_{\Pi_{\mathsf{Com}}}(\mathcal{A}) := \left| \Pr \left[\begin{array}{c} \mathsf{crs} \xleftarrow{\$} \{0,1\}^\ell, (\mathsf{M}_0, \mathsf{M}_1) \xleftarrow{\$} \mathcal{A}(\mathsf{crs}), b \xleftarrow{\$} \{0,1\} \\ \mathsf{com} \xleftarrow{\$} \mathsf{Com}(\mathsf{crs}, \mathsf{M}_b), b' \xleftarrow{\$} \mathcal{A}(\mathsf{crs}, \mathsf{com}) \end{array} : b = b' \right] - \frac{1}{2} \right| = \mathsf{negl}(\lambda).$$

2.5 Quantum Related Tools

Quantum Computation. We briefly give some background on quantum computation. We refer to [NC00] for more details. A state $|\psi\rangle$ of n qubits is expressed as $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle \in \mathbb{C}^{2^n}$ where $\{\alpha_x\}_{x\in\{0,1\}^n}$ is a set of complex numbers such that $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$ and $\{|x\rangle\}_{x\in\{0,1\}^n}$ is an orthonormal basis on $\mathbb{C}^{2^n}$ (which is called a computational basis). If we measure $|\psi\rangle$ in the computational basis, then the outcome is a classical bit string $x \in \{0,1\}^n$ with probability $|\alpha_x|^2$, and the state becomes $|x\rangle$. The evolution of a quantum state can be described by a unitary matrix U, which transforms $|x\rangle$ into $U|x\rangle$. A quantum algorithm is composed of quantum evolutions described by unitary matrices and measurements. We also consider a quantum oracle algorithm, which can quantumly access to certain oracles. The running time $\mathsf{Time}(\mathcal{A})$ of a quantum algorithm $\mathcal{A}$ is defined to be the number of universal gates (e.g., Hadamard, phase, CNOT, and $\pi/8$ gates) and measurements required for running $\mathcal{A}$.

Useful lemmata. Zhandry [Zha12] has shown that a quantum random oracle can be simulated by a family of $2\mathsf{Q}$-wise independent hash functions against an adversary that quantumly accesses the oracle at most Q times.

Lemma 2.23. *Any quantum algorithm* $\mathcal{A}$ *making quantum queries to random oracles can be efficiently simulated by a quantum algorithm* $\mathcal{B}$*, which has the same output distribution, but makes no queries. Especially, if* $\mathcal{A}$ *makes at most* Q *queries to a random oracle* $\mathsf{H} : \{0,1\}^a \to \{0,1\}^b$*, then* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{Q} \cdot T^{2\mathsf{Q}\text{-wise}}_{a,b}$ *where* $T^{2\mathsf{Q}\text{-wise}}_{a,b}$ *denotes the time to evaluate a* $2\mathsf{Q}$*-wise independent hash function from* $\{0,1\}^a$ *to* $\{0,1\}^b$*.*

Throughout the paper, we omit the subscripts a and b when the context is clear. The following two lemmata by Zhandry [Zha12] roughly states that we can modify the random oracle to have range with size polynomially related to the number of (quantum) random oracle query an adversary performs.

Definition 2.24 (Small-Range Distributions). *Fix a positive integer* r *and sets* $\mathcal{X}$ *and* $\mathcal{Y}$ *and a distribution* D *on* $\mathcal{Y}$*. Let* $\mathbf{y} = (y_1, \cdots, y_r) \xleftarrow{\$} D^r$ *and let* $P : \mathcal{X} \to [r]$ *be a random function. We define a* small-range distribution *with* r *samples of* D *by the distribution on* $\mathsf{Func}(\mathcal{X}, \mathcal{Y})$ *induced by* $\mathbf{y}$ *and* P *defined as* $\mathsf{H}(x) = y_{P(x)}$*.*

Lemma 2.25. *There is a universal constant* $C_0 = (8\pi^2)/3 \leq 27$ *such that, for any sets* $\mathcal{X}$ *and* $\mathcal{Y}$*, distribution* D *on* $\mathcal{Y}$*, any positive integer* r*, and any quantum algorithm* $\mathcal{A}$ *making* Q *queries to an oracle* $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$*, the following two cases are indistinguishable, except with probability less than* $C_0 \cdot \mathsf{Q}^3/r$*:*

- $\mathsf{H}(x) = y_{\mathsf{ind}(x)}$*, where* $\mathbf{y} = (y_1, \cdots, y_{|\mathcal{X}|}) \xleftarrow{\$} D^{|\mathcal{X}|}$ *and* ind *is any bijective map from* $\mathcal{X}$ *to* $[|\mathcal{X}|]$*;*
- H *is drawn from the small-range distribution with* r *samples of* D*.*

The following lemma by Zhandry [Zha12] states that if each output of two oracles are independent and computationally indistinguishable, then an efficient adversary with quantum access to the oracles can still not distinguish them,

Lemma 2.26. *Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary sets and let D_0 and D_1 be efficiently sampleable distributions on $\mathcal{Y}$. For $b \in \{0,1\}$, let $\mathcal{H}_b$ be a distribution over $\mathsf{Func}(\mathcal{X},\mathcal{Y})$ such that when we take $\mathsf{H}_b \leftarrow \mathcal{H}_b$, for each $x \in \mathcal{X}$, $\mathsf{H}_b(x)$ is identically and independently distributed according to D_b. Then if $\mathcal{A}$ is a QPT algorithm that makes at most Q oracle queries such that*

$$\left|\Pr[\mathcal{A}^{|\mathsf{H}_0\rangle}(1^\lambda) \to 1] - \Pr[\mathcal{A}^{|\mathsf{H}_1\rangle}(1^\lambda) \to 1]\right| \geq \epsilon,$$

where $\mathsf{H}_b \leftarrow \mathcal{H}_b$ for $b \in \{0,1\}$, then we can construct a QPT algorithm $\mathcal{B}$ with runtime similar to $\mathcal{A}$ that distinguishes D_0 from D_1 with probability at least $\epsilon^2/(C \cdot \mathsf{Q}^3)$ for some universal constant $C > 0$.

Finally, Kiltz, Lyubashevsky, and Schaffner [KLS18, Lemma 2.1] establishes that it is difficult even for an adversary with quantum access to the random oracle to find an input that satisfies a sparse relation. For any $\lambda \in [0,1]$, let $\mathcal{B}_\lambda$ denote the Bernoulli distribution, i.e., $\Pr_{b \leftarrow B_\lambda}[b = 1] = \lambda$.

Lemma 2.27 (Generic Search Problem with Bounded Probabilities). *Let $\lambda \in [0,1]$ and X be any set. For any (possibly unbounded) quantum algorithm $\mathcal{A}$ making at most Q quantum queries to its oracle, consider the following game between a challenger:*

1. *$\mathcal{A}$ outputs a set of reals $(\lambda_x)_{x \in X}$;*
2. *The challenger checks if $\lambda_x \leq \lambda$ for all $x \in X$. If not, abort. Otherwise, it samples $b_x \leftarrow \mathcal{B}_{\lambda_x}$ and prepares the function $\mathsf{G} : X \to \{0,1\}$ such that $\mathsf{G}(x) = b_x$ for all $x \in X$, and finally provides $\mathcal{A}$ oracle access to G;*
3. *$\mathcal{A}^{|\mathsf{G}\rangle}$ outputs $x \in X$. We say $\mathcal{A}$ wins if $\mathsf{G}(x) = 1$.*

Then, we have $\mathsf{Adv}^{\mathsf{GSBP}}(\mathcal{A}) := \Pr[\mathcal{A} \text{ wins}] \leq 8 \cdot \lambda \cdot (\mathsf{Q}+1)^2$.

3 Lattice-based Blind Signature from Compatible Commitments

In this section, we provide our generic construction of a blind signature tailored to lattices. For a high level overview of our construction, we refer the readers to Section 1.3. For simplicity, we first prove the scheme against classical adversaries. The proof against quantum adversaries is provided in Section 5.1.

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. [BDL+18] 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 $\mathsf{crs} \in \{0,1\}^{\ell_{\mathsf{com}}}$, $\mathsf{rand} \in \mathcal{R}$, $\vec{\mathsf{M}} \in \mathcal{M}$, and $\mathsf{com} = \mathsf{Com}(\mathsf{crs}, \vec{\mathsf{M}}; \mathsf{rand})$, we have the following:*

- *$(\mathbf{b}_i)_{i\in[L]} \subseteq \mathsf{crs}$[9], $\mathbf{t} = \mathsf{ParseCom}(\mathsf{com})$, 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 $\vec{\mathsf{M}}$, and $\mathbf{r}_i$ satisfies $s_1\big([\mathbf{r}_1^\top | \dots | \mathbf{r}_L^\top]\big) \leq \delta$;*

[9] 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 $(R_q^k)^L \times \{0,1\}^\ell$. Although we could have considered an algorithm $\mathsf{ParseCRS}$ that outputs $(\mathbf{b}_i)_{i\in[L]}$ on input crs, we did not chose so since it would complicate the security proof. (See Footnote 12)

- *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 $s_1(\cdot)$ 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. We require these parameters to satisfy certain conditions for the correctness and security to hold, which are summarized in Section 3.3. As typical with many lattice-based constructions, the parameters are quite dense so we advise the readers to refer Table 1 only when needed.

Looking ahead, 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}. We can simply set k_1, k_2, and k_3 to be equal to the maximum value of these three but we chose to parameterize them since it allows us to fine-tune them for better concrete efficiency. For those only interested in the asymptotic, one can safely assume they are the same value.

Parameter	Explanation
R_q	Polynomial ring $R_q = \mathbb{Z}[X]/(q, X^d + 1)$
B_{inv}	Any $a \in R_q$ s.t. $\|a\|_2 \le B_{\mathsf{inv}}$ is invertible
k_1	Size of lattice trapdoor $\mathbf{T} \in R^{k_1 \times k_1}$ (see Lemmata 2.19 and 2.20)
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} (see Definition 3.1)
σ	Gaussian parameter for trapdoor sampling algorithms
$(\ell^{\mathsf{m}}_{\mathsf{NIZK}}, \ell_{\mathsf{com}})$	Length of common random string crs for $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ and Π_{Com}, respectively
δ^{gap}	Spectral norm bound on the extracted commitment randomness used in gap relation $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$
$B^{\mathcal{S}}_{\Sigma,i},\ i \in [3]$	Two-norm bound on the vector $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) := \mathbf{e}$ sampled by the signer
$B^{\mathcal{U}}_{\Sigma,i},\ i \in [3]$	Two-norm bound on the real secret vector $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}}$ for relation $\mathcal{R}^{\mathsf{s}}$
$B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i},\ i \in [3]$	Two-norm bound on the extracted vector $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}}$ for gap relation $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$
$S_{\mathsf{chal}} \subset R_q$	Challenge set of the interactive proof system implicit in $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$
B_c	One-norm bound on $c \in S_{\mathsf{chal}}$ used in gap relation $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$
$S_{\mathsf{hash}} \subset R_q$	Hashed message set with size $> 2^\lambda$ s.t. $\forall (c,h) \in S_{\mathsf{chal}} \times S_{\mathsf{hash}}$, $\|c \cdot h\|_2 \le B_{\mathsf{inv}}/2$
Δ_{MLWE}	Bound on solution size of *search* MLWE s.t. the solution is not unique
$(\chi_{\mathsf{MLWE}}, B_{\mathsf{MLWE}})$	Noise distribution for *decision* MLWE, where $\mathbf{R} \xleftarrow{\$} \chi^{k_1 \times k_2}_{\mathsf{MLWE}} \Rightarrow s_1(\mathbf{R}) \le B_{\mathsf{MLWE}}$ 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.

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 Definition 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:[10]
  - $\mathcal{R}^{\mathsf{s}} := \left\{ \mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h), \mathsf{W} = \tilde{\mathbf{e}} \;\middle|\; \begin{array}{c} (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}} \in R^{k_1+k_2+k_2\cdot k_3}, \\ \forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U}}_{\Sigma,i} \\ \wedge\ [\mathbf{a}_1 \mid \mathbf{a}_2 + h\cdot\mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}]\,\tilde{\mathbf{e}}^\top = u \end{array} \right\};$
  - $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}} := \left\{ \mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h), \mathsf{W} = (\tilde{\mathbf{e}}, c) \;\middle|\; \begin{array}{c} (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}} \in R^{k_1+k_2+k_2\cdot k_3}, \\ \forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i} \ \wedge\ \|c\|_1 \le B_c \\ \wedge\ [\mathbf{a}_1 \mid \mathbf{a}_2 + h\cdot\mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}]\,\tilde{\mathbf{e}}^\top = c\cdot u \end{array} \right\}.$
- 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:
  - $\mathcal{R}^{\mathsf{m}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \\ \mathsf{W} = (h, \mathsf{rand}) \end{array} \;\middle|\; \begin{array}{c} (h, \mathsf{rand}) \in S_{\mathsf{hash}} \times \mathcal{R}, \\ \wedge\ \mathsf{com} = \mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, h\cdot\mathbf{g}; \mathsf{rand}) \end{array} \right\};$
  - $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \\ \mathsf{W} = (h', c', c, (\mathbf{r}_i)_{i\in[k_2]}) \end{array} \;\middle|\; \begin{array}{c} \|h'\|_2 \le B_{\mathsf{inv}}/2 \ \wedge\ \|c'\|_1, \|c\|_1 \le B_c \\ \wedge\ s_1\big([\mathbf{r}_1^\top | \cdots | \mathbf{r}_{k_2}^\top]\big) \le \delta^{\mathsf{gap}} \\ \wedge\ t_i = \mathbf{b}_i(\mathbf{r}_i/c)^\top + (h'/c')\cdot g_i \end{array} \right\},$

  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 $\mathbf{a}_2 \in R_q^{k_2}$. Note that as standard practice, the four hash functions can be derived from a single hash function by using appropriate domain separation.

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 $\mathbf{s} \xleftarrow{\$} [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]^{(k_1+k_2k_3)}_{\mathsf{coeff}}$ 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 $\rho_2 = \mathbf{e}$.

[10] 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 $(\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) \in R^{k_1} \times R^{k_2} \times R^{k_2\cdot k_3}$.

$\mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2)$: 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 intro 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.

$\mathsf{BSVerify}(\mathsf{vk}, \mathsf{M}, \Sigma)$: 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 3.2 (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 Section 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). The user then parses

$$[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1] \cdot \mathbf{e}^\top = [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \\ [\mathbf{r}_1^\top \mid \cdots \mid \mathbf{r}_{k_2}^\top]\, \mathbf{e}_2^\top + \mathbf{e}_3^\top \end{bmatrix},$$

where it reconstructs a vector with a slightly larger norm but shorter dimension compared to $\tilde{\mathbf{e}}$ defined above. 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. However, regardless of which variation is chosen, the following security proofs we provide remains identical.

3.3 Correctness and Condition on Parameters

Correctness. The following establishes the correctness of the above blind signature Π_{BS}.

Lemma 3.3. *The blind signature Π_{BS} is correct if $\sigma > \omega(q^{1/k_1} \cdot \sqrt{\log k_1 d})$, $\forall i \in [3], B^{\mathcal{S}}_{\Sigma,i} = \sqrt{k_i d}\sigma$, $\forall i \in [2], B^{\mathcal{U}}_{\Sigma,i} = B^{\mathcal{S}}_{\Sigma,i}, B^{\mathcal{U}}_{\Sigma,3} = \delta B^{\mathcal{S}}_{\Sigma,2} + B^{\mathcal{S}}_{\Sigma,3}$ and the two NIZKs $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ are correct.*

Proof. By correctness of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$, the signer correctly processes the first message ρ_1 sent from the user. By Lemma 2.19, we have $\|\mathbf{T}_{\mathbf{a}_1}\|_{\mathsf{GS}} = O(q^{1/k_1})$. Combining this with Lemmata 2.14 and 2.20 and the bound on the Gaussian parameter σ, the samples vector $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) := \mathbf{e}$ satisfies $\|\mathbf{e}_i\|_2 \le B^{\mathcal{S}}_{\Sigma,i}$ for all $i \in [3]$ with all but negligible probability. Then, we have $\|\tilde{\mathbf{e}}_3\|_2 \le \|\mathbf{e}_3\|_2 + \sum_{i=1}^{k_2} \|e_{2,i} \cdot \mathbf{r}_i\|_2 \le \|\mathbf{e}_3\|_2 + \|\mathbf{e}_2\|_2 \cdot s_1([\mathbf{r}_1^\top | \ldots | \mathbf{r}_{k_2}^\top]) \le \delta B^{\mathcal{S}}_{\Sigma,2} + B^{\mathcal{S}}_{\Sigma,3}$, where we use $s_1([\mathbf{r}_1^\top | \ldots | \mathbf{r}_{k_2}^\top]) \le \delta$ which follows from the (k_3, δ)-trapdoor-sampling-compatibility of Π_{Com}. Hence, by correctness of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$, we conclude that Π_{BS} is correct with all but negligible probability. □

Conditions on Parameters. We summarize the conditions that our parameters in Table 1 must satisfy for the correctness and security of our scheme. These conditions are only asymptotic and mainly provided for concreteness. We show in Section 4.5 a set of concrete parameters.

- The $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$, $\mathsf{DSMR}_{d,k_1-1,\chi_{\mathsf{DSMR}},q,1}$, and $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ assumptions hold, where for any $\mathbf{R} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1 \times k_2}$, we have $s_1(\mathbf{R}) \leq B_{\mathsf{MLWE}}$ with overwhelming probability;
- The $\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}$ assumption with $B_{\mathsf{MSIS}} = B_{\Sigma,1}^{\mathcal{U},gap} + B_{\Sigma,3}^{\mathcal{U},gap} + B_{\mathsf{MLWE}} \cdot B_{\Sigma,2}^{\mathcal{U},gap} + B_c \cdot \Delta_{\mathsf{MLWE}} \sqrt{k_1 + k_2 k_3}$ holds;
- $\mathsf{TrapGen}$ operates properly (Lemma 2.19): that is, $B_{\mathsf{DSMR}} \lesssim O(q^{1/k_1})$.
- $\mathsf{SampleLeft}$ operates properly (Lemma 2.20): that is, $\sigma > \omega(q^{1/k_1} \cdot \sqrt{\log k_1 d})$;
- $\mathsf{SampleRight}$ (in the security proof) operates properly (Lemma 2.20): that is, $\sigma > s_1(\mathbf{R}') \cdot \|\mathbf{T_g}\|_{\mathsf{GS}} \cdot \omega(\sqrt{\log k_2 d})$, where $s_1(\mathbf{R}')^2 \leq \sqrt{B_c \cdot B_{\mathsf{MLWE}}^2 + \delta^{\mathsf{gap}^2}}$, and $\|\mathbf{T_g}\|_{\mathsf{GS}} \leq O(q^{1/k_2})$;
- Vector $\mathbf{s}$ sampled by BSGen retains 1-bit of min-entropy (Lemma 2.15): that is, $(2q)^{1/(k_1+k_2k_3)}/2 \leq \Delta_{\mathsf{MLWE}}$;
- Correctness holds: that is $B_{\Sigma,i}^{\mathcal{S}} = \sqrt{k_i d}\sigma$, $\forall i \in [2], B_{\Sigma,i}^{\mathcal{U}} = B_{\Sigma,i}^{\mathcal{S}}, B_{\Sigma,3}^{\mathcal{U}} = \delta B_{\Sigma,2}^{\mathcal{S}} + B_{\Sigma,3}^{\mathcal{S}}$;
- $B_{\Sigma,i}^{\mathcal{U}} \leq B_{\Sigma,i}^{\mathcal{U},\mathsf{gap}}$ and $\delta \leq \delta^{\mathsf{gap}}$ that are required implicitly by the extractability of $\Pi_{\mathsf{NIZK}}^{\mathsf{s}}$ and $\Pi_{\mathsf{NIZK}}^{\mathsf{m}}$, respectively;
- Condition on $(B_{\mathsf{inv}}, S_{\mathsf{hash}}, S_{\mathsf{chal}})$ holds: that is, any $a \in R_q$ s.t. $\|a\|_2 \leq B_{\mathsf{inv}}$ is invertible, $|S_{\mathsf{hash}}| \geq 2^\lambda$, and for any $(c, h) \in S_{\mathsf{chal}} \times S_{\mathsf{hash}}$, we have $\|c \cdot h\|_2 \leq B_{\mathsf{inv}}/2$.

3.4 Proof of Blindness

Theorem 3.4. *The blind signature Π_{BS} is classically blind under malicious keys if the commitment scheme Π_{Com} is classically hiding, and the two NIZKs $\Pi_{\mathsf{NIZK}}^{\mathsf{s}}$ for $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}_{\mathsf{gap}}^{\mathsf{s}})$ and $\Pi_{\mathsf{NIZK}}^{\mathsf{m}}$ for $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}_{\mathsf{gap}}^{\mathsf{m}})$ are classically zero-knowledge.*

Proof. Let $\mathcal{A}$ be a PPT adversary against the blindness game. Below, we consider a sequence of games, where the challenger samples $\mathsf{coin} = 0$ (resp. 1) in the first (resp. last) game. For each i, let ϵ_i denote the probability that $\mathcal{A}$ outputs $\mathsf{coin}' = 0$ in Game_i. Blindness is established by showing that the differences between the ϵ_i in each adjacent games are negligible.

Game_1 : This is the real blindness game where the challenger samples $\mathsf{coin} = 0$. Specifically, $(\rho_{1,0}, \rho_{1,1})$ is given to $\mathcal{A}$ as the first message. By definition, $\mathcal{A}$ outputs $\mathsf{coin}' = 0$ with probability ϵ_1.

Game_2 : In this game, instead of running $\mathsf{Prove}^{\mathsf{H_s}}$, the challenger simulates the signature using the zero-knowledge simulator $\mathsf{Sim}^{\mathsf{s}} = (\mathsf{Sim}_0^{\mathsf{s}}, \mathsf{Sim}_1^{\mathsf{s}})$ for $\Pi_{\mathsf{NIZK}}^{\mathsf{s}}$. Concretely, when $\mathcal{A}$ makes a random oracle query to $\mathsf{H_s}$ the challenger runs $\mathsf{Sim}_0^{\mathsf{s}}$. When $\mathcal{A}$ submits $(\rho_{2,0}, \rho_{2,1})$ to the challenger, the challenger parses $\mathbf{e}_{2,b} \leftarrow \rho_{2,b}$ for $b \in \{0,1\}$ and performs the check made by $\mathcal{U}_{\mathsf{der}}$. If it holds, it runs $\tilde{\pi}_b^{\mathsf{s}} \xleftarrow{\$} \mathsf{Sim}_1^{\mathsf{s}}(\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h_b)$ for $b \in \{0,1\}$, where $h_b = \mathsf{H_M}(\mathsf{M}_b)$. If the two simulated proofs $\tilde{\pi}^{\mathsf{s}}$ are valid, then it outputs the two signatures as $(\Sigma_0 := \tilde{\pi}_0^{\mathsf{s}}, \Sigma_1 := \tilde{\pi}_1^{\mathsf{s}})$ to $\mathcal{A}$. Notice that by definition of the blindness game, the challenger runs $\mathsf{Sim}_1^{\mathsf{s}}$ at most twice per statement as required by the definition of $\mathsf{Sim}_1^{\mathsf{s}}$.[11]

It can be checked that we can construct a PPT adversary $\mathcal{B}^{\mathsf{s}}$ that has advantage $|\epsilon_1 - \epsilon_2|$ in the zero-knowledge game, where $\mathcal{B}^{\mathsf{s}}$ internally executes $\mathcal{A}$ and simulates the challenger with its provided oracles $(\mathcal{O}, \mathsf{Prove})$ or $(\mathsf{Sim}_0, \mathcal{S})$. Note that $\mathcal{B}^{\mathsf{s}}$ only queries valid statements to Prove or $\mathcal{S}$ due to the check performed by $\mathcal{U}_{\mathsf{der}}$. Moreover, $\mathcal{B}^{\mathsf{s}}$ can answer the random oracle queries to $\mathsf{H_{crs}}, \mathsf{H_M}$, and $\mathsf{H_m}$ in an on-the-fly manner. Thus we have,

$$|\epsilon_1 - \epsilon_2| \leq \mathsf{Adv}_{\Pi_{\mathsf{NIZK}}^{\mathsf{s}}}^{\mathsf{ZK}}(\mathcal{B}^{\mathsf{s}}).$$

[11] In standard proof of Fiat-Shamir-based signatures, this subtle condition is typically ignored since we can turn the signing algorithm deterministic using a pseudorandom function. That is, $\mathsf{Sim}_1^{\mathsf{s}}$ only needs to generate one proof per statement. In the context of blind signatures, this is no longer the case since even if two users use the same statement, the proofs generated with different randomness will be different.

Game_3 : In this game, instead of running $\mathsf{Prove}^{\mathsf{H}_\mathsf{m}}$, the challenger modifies part of the first message ρ_1 using the zero-knowledge simulator $\mathsf{Sim}^\mathsf{m} = (\mathsf{Sim}^\mathsf{m}_0, \mathsf{Sim}^\mathsf{m}_1)$ for $\Pi^\mathsf{m}_{\mathsf{NIZK}}$. In particular, when $\mathcal{A}$ makes a random oracle query to H_m the challenger runs $\mathsf{Sim}^\mathsf{m}_0$. Moreover, when $\mathcal{A}$ submits $(\mathsf{M}_0, \mathsf{M}_1)$ to the challenger, the challenger computes $\mathsf{com}_b \xleftarrow{\$} \mathsf{Com}(\mathsf{crs}_\mathsf{com}, h_b \cdot \mathbf{g})$ for $b \in \{0,1\}$, where $h_b = \mathsf{H}_\mathsf{M}(\mathsf{M}_b)$ as performed by $\mathcal{U}_1$. It then runs $\tilde{\pi}^\mathsf{m}_b \xleftarrow{\$} \mathsf{Sim}^\mathsf{m}_1(\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}, (\mathsf{crs}_\mathsf{com}, \mathsf{com}_b))$ for $b \in \{0,1\}$, and outputs the first message pairs as $(\rho_{1,0} := (\mathsf{com}_0, \tilde{\pi}^\mathsf{m}_0), \rho_{1,1} := (\mathsf{com}_1, \tilde{\pi}^\mathsf{m}_1))$ to $\mathcal{A}$.

Following an identically argument to above and further programming the output of $\mathsf{H}_\mathsf{crs}(0)$ to use $\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}$ provided by the zero-knowledge game of $\Pi^\mathsf{m}_{\mathsf{NIZK}}$, we can construct a PPT adversary $\mathcal{B}^\mathsf{m}$ such that

$$|\epsilon_2 - \epsilon_3| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\Pi^\mathsf{m}_{\mathsf{NIZK}}}(\mathcal{B}^\mathsf{m}).$$

Game_4 : In this game, the challenger further modifies part of the first message ρ_1. Rather than computing $\mathsf{com}_b \xleftarrow{\$} \mathsf{Com}(\mathsf{crs}_\mathsf{com}, h_b \cdot \mathbf{g})$ for $b \in \{0,1\}$, the challenger computes $\mathsf{com}_b \xleftarrow{\$} \mathsf{Com}(\mathsf{crs}_\mathsf{com}, \mathbf{0})$ for $b \in \{0,1\}$, where $\mathbf{0} \in \mathcal{M} = R_q^{k_2}$. By programming the output of $\mathsf{H}_\mathsf{crs}(0)$ to use $\mathsf{crs}_\mathsf{com}$ provided by the hiding game of Π_Com, it is clear that we can construct a PPT adversary $\mathcal{B}_\mathsf{com}$ such that

$$|\epsilon_3 - \epsilon_4| \leq 2 \cdot \mathsf{Adv}^{\mathsf{hide}}_{\Pi_\mathsf{Com}}(\mathcal{B}_\mathsf{com}).$$

At this point, the distribution of the first messages $(\rho_{1,0}, \rho_{1,1})$ and signatures (Σ_0, Σ_1) given to $\mathcal{A}$ are independent of the distribution coin sampled by the challenger. In other words, the adversaries advantage remains the same even if the challenger sends $(\rho_{1,1}, \rho_{1,0})$ as the first message.

Game_5 : This is the real blindness game where the challenger samples $\mathsf{coin} = 1$ and $(\rho_{1,1}, \rho_{1,0})$ is given to $\mathcal{A}$ as the first message. By redoing the modifications made to move from Game_1 to Game_4 in reverse order, while setting $\mathsf{coin} = 1$, we have $|\epsilon_4 - \epsilon_5| = |\epsilon_1 - \epsilon_4|$.

Collecting all the bounds, we have $|\epsilon_1 - \epsilon_5| = \mathsf{negl}(\lambda)$ as desired. Moreover, $\mathsf{Time}(\mathcal{B}^\mathsf{s}), \mathsf{Time}(\mathcal{B}^\mathsf{m})$, and $\mathsf{Time}(\mathcal{B}^{\mathsf{com}})$ are roughly the same as $\mathsf{Time}(\mathcal{A})$. □

3.5 Proof of One-More Unforgeability

Theorem 3.5. *The blind signature Π_{BS} is classically one-more unforgeable if the two NIZKs $\Pi^\mathsf{s}_{\mathsf{NIZK}}$ 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 classically 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. Assume there exists a PPT adversary $\mathcal{A}$ with non-negligible advantage ϵ against the one-more unforgeability game that makes at most Q_S signature queries. Further assume $\mathcal{A}$ makes at most $\mathsf{Q}_{\mathsf{H}_\mathsf{M}}$ (resp. $\mathsf{Q}_{\mathsf{H}_\mathsf{m}}, \mathsf{Q}_{\mathsf{H}_\mathsf{s}}$) random oracle queries to H_M (resp. H_m, H_s), where we assume $\mathcal{A}$ never repeats the same query without loss of generality. We consider a sequence of games, where we denote E_i as the event $\mathcal{A}$ wins in Game_i and $\mathcal{C}_i$ as the challenger in Game_i.

Game_1 : This is the real one-more unforgeability game. By definition, we have

$$\Pr[\mathsf{E}_1] = \epsilon.$$

Game_2 : In this game, the challenger modifies the output of $\mathsf{H}_\mathsf{crs}(0) = (\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}, \mathsf{crs}_\mathsf{com}, \mathbf{a}_2)$. In the previous game, $\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}} \xleftarrow{\$} \{0,1\}^{\ell^\mathsf{m}_{\mathsf{NIZK}}}$. In this game, the challenger runs the CRS simulator $\mathcal{S}_\mathsf{crs}$ provided by $\Pi^\mathsf{m}_{\mathsf{NIZK}}$ and generates $(\widetilde{\mathsf{crs}}^\mathsf{m}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_\mathsf{crs}(1^\lambda)$. It programs $\mathsf{H}_\mathsf{crs}(0)$ to output $\widetilde{\mathsf{crs}}^\mathsf{m}_{\mathsf{NIZK}}$ instead of $\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}$. Otherwise, it proceeds identically to Game_1.

It can be checked that Game_1 and Game_2 are indistinguishable by the CRS indistinguishability in Definition 2.10. Specifically, there exists a PPT adversary $\mathcal{B}_{\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}}$ against the CRS indistinguishability such that

$$\Pr[\mathsf{E}_2] \geq \Pr[\mathsf{E}_1] - \mathsf{Adv}^{\mathsf{crs}}_{\Pi^\mathsf{m}_{\mathsf{NIZK}}}(\mathcal{B}_{\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}}),$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}})$ is $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_2)$, which is roughly $\mathsf{Time}(\mathcal{A})$.

Game_3 : In this game, the challenger uses the multi-proof extractor $\mathsf{Multi\text{-}Extract}$ provided by $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ to extract a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ from all the proofs included in Q_{S} first messages $(\rho_{j,1})_{j\in[\mathsf{Q}_{\mathsf{S}}]}$ submitted by $\mathcal{A}$. Specifically, when $\mathcal{A}$ submits $\rho_{j,1} = (\mathsf{com}_j, \pi^{\mathsf{m}}_j)$ to the challenger, 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 $\mu = \Pr[\mathsf{E}_2]$ and $\mathsf{X}_j = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}_j)$. We denote by $\mathsf{Abort}_{\mathsf{extract}}$ the event that there exists $j \in [\mathsf{Q}_{\mathsf{S}}]$ such that $\mathsf{W}_j \notin \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. If $\mathsf{Abort}_{\mathsf{extract}}$ occurs, the challenger aborts the game and rewrites the forgery of $\mathcal{A}$ to be $\perp$. Otherwise, it proceeds identically to Game_2. Conditioning on $\mathsf{Abort}_{\mathsf{extract}}$ not occurring, the challenger extracts $\mathsf{W}_j = (h'_j, c'_j, c_j, (\mathbf{r}_{j,i})_{i\in[k_2]}) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. We note that the challenger does not use the extracted witness in this game.

We later show in Lemma 3.6 that

$$\Pr[\mathsf{E}_3] \geq \frac{1}{2} \cdot \Pr[\mathsf{E}_2] - \mathsf{negl}(\lambda).$$

Note that the runtime of the challenger $\mathcal{C}_3$ becomes longer than that of $\mathcal{C}_2$ since it runs the multi-proof extractor $\mathsf{Multi\text{-}Extract}$. Due to Definition 2.10, we have $\mathsf{Time}(\mathcal{C}_3) = \mathsf{Time}(\mathcal{C}_2) + \mathsf{Q}^{e_1}_{\mathsf{H}_{\mathsf{m}}} \cdot \mathsf{Q}^{e_2+1}_{\mathsf{S}} \cdot \frac{1}{\mu^c} \cdot p(\lambda)$ for some constants (c, e_1, e_2) and polynomial $p(\lambda)$, where $\mu = \Pr[\mathsf{E}_2] \geq \epsilon - \mathsf{negl}(\lambda)$. Assuming ϵ is non-negligible, $\mathsf{Time}(\mathcal{C}_3)$ is bounded by a polynomial.

Game_4 : In this game, the challenger guesses the timing on which one of the messages included in the forgery output by $\mathcal{A}$ is queried to the random oracle H_{M}. Specifically, at the beginning of the game, the challenger samples $j^* \xleftarrow{\$} [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$ and $h_j \xleftarrow{\$} S_{\mathsf{hash}}$ for all $j \in [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$. When $\mathcal{A}$ queries M'_j as its j-th ($j \in [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$) random oracle query to H_{M}, the challenger simply returns h_j. The challenger performs two types of checks. First, when the challenger extracts $\mathsf{W}_j = (h'_j, c'_j, c_j, (\mathbf{r}_{j,i})_{i\in[k_2]}) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ from the first message $\rho_{j,1}$ submitted to by $\mathcal{A}$ (conditioned on $\mathsf{Abort}_{\mathsf{extract}}$ not occurring), the challenger checks if $h'_j/c'_j \neq h_{j^*}$, where note that by definition c'_j is invertible. 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. We denote by $\mathsf{Abort}_{\mathsf{guess}}$ the event that either of these checks do not hold. If $\mathsf{Abort}_{\mathsf{guess}}$ occurs, the challenger aborts and rewrites the forgery of $\mathcal{A}$ to be $\perp$. Otherwise, it proceeds identically to Game_3.

We later show in Lemma 3.7 that

$$\Pr[\mathsf{E}_4] \geq \frac{1}{\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}} \cdot \left(\Pr[\mathsf{E}_3] - \frac{\mathsf{Q}^2_{\mathsf{H}_{\mathsf{M}}} + 1}{|S_{\mathsf{hash}}|} \right).$$

Game_5 : In this game, the challenger modifies the output of $\mathsf{H}_{\mathsf{crs}}(0) = (\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2)$. Specifically, after it samples $j^* \xleftarrow{\$} [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$ and $h_j \xleftarrow{\$} S_{\mathsf{hash}}$ for all $j \in [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$ at the beginning of the game, it sets $\mathbf{a}_2 = \widetilde{\mathbf{a}}_2 - h_{j^*} \cdot \mathbf{g}$ where $\widetilde{\mathbf{a}}_2 \xleftarrow{\$} R^{k_2}_q$. It then programs $\mathsf{H}_{\mathsf{crs}}(0)$ to output this $\mathbf{a}_2$ rather than $\mathbf{a}_2 \xleftarrow{\$} R^{k_2}_q$ as in the previous game. It is clear that the distribution of both $\mathbf{a}_2$ are identical. Thus, we have

$$\Pr[\mathsf{E}_5] = \Pr[\mathsf{E}_4].$$

Game_6 : In this game, the challenger gets rid of the trapdoor $\mathbf{T}_{\mathbf{a}_1}$ included in the secret key sk and modifies the way it samples the short vector $\mathbf{e}$ when $\mathcal{A}$ submits the first message ρ_1. In particular, the challenger modifies the output of $\mathsf{H}_{\mathsf{crs}}(0)$ and the two algorithms BSGen and $\mathcal{S}_2$ as follows, where the modification from the previous game is underlined in red.

$\mathsf{H}_{\mathsf{crs}}(0)$: It sample $(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^\lambda)$ and $\underline{\mathbf{R} \xleftarrow{\$} \chi^{k_1\times k_2}_{\mathsf{MLWE}}}$ and sets $\underline{\widetilde{\mathbf{a}}_2 = \mathbf{a}_1\mathbf{R}}$, where $\mathbf{a}_1$ is defined in BSGen below. It then sets the output of the random oracle to be $(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2 = \widetilde{\mathbf{a}}_2 - h_{j^*} \cdot \mathbf{g})$,

$\mathsf{BSGen}(1^\lambda)$: It samples $\underline{\mathbf{a}_1 \xleftarrow{\$} R^{k_1}_q}$, $\mathbf{s} \xleftarrow{\$} [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]^{k_1+k_2+k_2k_3}_{\mathsf{coeff}}$ and sets $u = [\mathbf{a}_1 \mid \mathbf{a}_2 \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \cdot \mathbf{s}^\top \in R_q$. It then outputs $(\mathsf{vk}, \underline{\widetilde{\mathsf{sk}}}) = ((\mathbf{a}_1, u), \underline{(\tau, \mathbf{R})})$.

$\mathcal{S}_2(\widetilde{\mathsf{sk}}, \rho_1)$: It parses $(\mathsf{com}, \pi^{\mathsf{m}}) \xleftarrow{\$} \rho_1$ and outputs $\perp$ if $\mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \pi^{\mathsf{m}}) = \perp$. Otherwise, it runs $\mathsf{W} \leftarrow \mathsf{Multi\text{-}Extract}(1^\lambda, \mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}, \mathsf{Q}_{\mathsf{S}}, 1/\mu, \tau, \mathsf{X}, \pi^{\mathsf{m}})$, where μ is defined as in Game_3 and $\mathsf{X} = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com})$. Conditioning on event $\mathsf{Abort}_{\mathsf{extract}}$ not occurring, we have $\mathsf{W} = (h', c', c, (\mathbf{r}_i)_{i\in[k_2]}) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. Recall that by definition of $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$, we have $t_i = \mathbf{b}_i(\mathbf{r}_i/c)^\top + (h'/c') \cdot \mathbf{g}_i$ for all $i \in [k_2]$, where $\mathbf{t} \leftarrow \mathsf{ParseCom}(\mathsf{com})$, $\|h'\|_2 \le B_{\mathsf{inv}}/2$, and c', c are guaranteed to be invertible and small, i.e., $\|c'\|_1, \|c\|_1 \le B_c$. It then rewrites the vector as follows:

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

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 $\mathbf{P}_{\mathsf{perm}} \in \{0,1\}^{(k_1+k_2+k_2k_3)\times(k_1+k_2+k_2k_3)}$ 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

$$
\left[\mathbf{a}_1 \mid \widehat{\mathbf{b}} \mid \left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right] \mathbf{R}' + \left(\frac{h'}{c'} - h_{j^*}\right) \cdot \mathbf{g}\right] \cdot \mathbf{e}'^\top = u, \tag{7}
$$

using $\underline{\mathbf{e}' \xleftarrow{\$} \mathsf{SampleRight}([\mathbf{a} \mid \widehat{\mathbf{b}}], \mathbf{g}, (\mathbf{R}, c, \widehat{\mathbf{R}}), (h'/c' - h_{j^*}), u, \mathbf{T}_{\mathbf{g}}, \sigma)}$, where note that invertibility of $\frac{h'}{c'} - h_{j^*}$ required by Lemma 2.20 can be checked as follows. First, we have $0 < \|h' - c' \cdot h_{j^*}\|_2 \le B_{\mathsf{inv}}$ based on our parameter selection (see Table 1 and $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$) and due to the condition that event $\mathsf{Abort}_{\mathsf{guess}}$ does not occur. This shows that $h' - c' \cdot h_{j^*}$ is invertible. Then, since $c' \cdot \left(\frac{h'}{c'} - h_{j^*}\right) = h' - c' \cdot h_{j^*}$ for an invertible c', $\frac{h'}{c'} - h_{j^*}$ must be invertible as well. The signer algorithm $\mathcal{S}_2$ finally outputs the second message $\rho_2 = \mathbf{e}'(\mathbf{P}_{\mathsf{perm}}^{-1})^\top$.

We later show in Lemma 3.8 that there exists PPT adversaries $\mathcal{B}_{\mathsf{MLWE}}$, $\mathcal{B}'_{\mathsf{DSMR}}$ and $\mathcal{B}_{\mathsf{DSMR}}$ against the $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$, $\mathsf{DSMR}_{d,k_1,\chi_{\mathsf{DSMR}},q,1}$, and $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ problems, respectively, such that

$$
\begin{aligned}
\Pr[\mathsf{E}_6] \ge \Pr[\mathsf{E}_5] - &\mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}}(\mathcal{B}_{\mathsf{MLWE}}) - \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}'_{\mathsf{DSMR}}) \\
&- 2 \cdot \mathsf{Adv}^{\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}_{\mathsf{DSMR}}) - \mathsf{negl}(\lambda)
\end{aligned}
$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{MLWE}})$, $\mathsf{Time}(\mathcal{B}'_{\mathsf{DSMR}})$, and $\mathsf{Time}(\mathcal{B}_{\mathsf{DSMR}})$ are roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$. Assuming the hardness of the MLWE and DSMR problems, we have $\Pr[\mathsf{E}_6] \ge \Pr[\mathsf{E}_5] - \mathsf{negl}(\lambda)$.

At this point, the challenger in Game_6 no longer relies on a trapdoor of $\mathbf{a}_1$. Therefore, we are now ready to embed an MSIS instance in the public vectors and to simulate the view of $\mathcal{A}$ in Game_6 in order to solve the MSIS problem. We formally show in Lemma 3.9 that there exists a PPT adversary $\mathcal{B}_{\mathsf{MSIS}}$ against the MSIS problem such that

$$
\mathsf{Adv}^{\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}}(\mathcal{B}_{\mathsf{MSIS}}) \ge \frac{1}{2p(\lambda) \cdot \mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}^e} \cdot \Pr[\mathsf{E}_6]^{c_1} - \mathsf{negl}(\lambda),
$$

where $p(\lambda)$ is a polynomial, and e and c_1 are constants defined in Definition 2.9. Moreover, we have $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \le c_2 \cdot (\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6))$, where c_2 is also a constant defined in Definition 2.9.

Let us check that $\mathcal{B}_{\mathsf{MSIS}}$ has non-negligible advantage and runs in polynomial time to arrive at a contradiction. Collecting all the bounds, we have

$$
\Pr[\mathsf{E}_6] \ge \frac{\Pr[\mathsf{E}_1]}{2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}} - \mathsf{negl}(\lambda) = \frac{\epsilon}{2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}} - \mathsf{negl}(\lambda),
$$

where we used $|S_{\mathsf{hash}}| \geq 2^\lambda$. This in particular implies

$$\mathsf{Adv}^{\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}}(\mathcal{B}_{\mathsf{MSIS}}) \geq \frac{1}{2p(\lambda)\cdot \mathsf{Q}_{\mathsf{H_s}}^e} \cdot \left(\frac{\epsilon}{2\mathsf{Q}_{\mathsf{H_M}}}\right)^{c_1} - \mathsf{negl}(\lambda)$$

which is non-negligible by assumption. Moreover, we have $\mathsf{Time}(\mathcal{C}_6) \approx \cdots \approx \mathsf{Time}(\mathcal{C}_3)$, $\mathsf{Time}(\mathcal{C}_3) = \mathsf{Time}(\mathcal{C}_2) + \frac{\mathsf{Q}_{\mathsf{H_m}}^{e_1}\cdot \mathsf{Q}_{\mathsf{S}}^{e_2+1}}{\mu^c}\cdot p(\lambda)$, and $\mathsf{Time}(\mathcal{C}_2) \approx \mathsf{Time}(\mathcal{C}_1)$, where "$\approx$" hides an insignificant blow up in the runtime and $\mu = \Pr[\mathsf{E}_2] \geq \epsilon - \mathsf{negl}(\lambda)$. Since $\mathsf{Time}(\mathcal{A})$ can be assumed to be larger than $\mathsf{Time}(\mathcal{C}_1)$, we have $\mathsf{Time}(\mathcal{C}_6) \approx \mathsf{Time}(\mathcal{A}) + \frac{\mathsf{Q}_{\mathsf{H_m}}^{e_1}\cdot \mathsf{Q}_{\mathsf{S}}^{e_2+1}}{\epsilon^c}\cdot p(\lambda)$, and thus, $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \lessapprox c_2 \cdot O\left(\mathsf{Time}(\mathcal{A}) + \frac{\mathsf{Q}_{\mathsf{H_m}}^{e_1}\cdot \mathsf{Q}_{\mathsf{S}}^{e_2+1}}{\epsilon^c}\cdot p(\lambda)\right)$. Since ϵ is non-negligible and $\mathsf{Time}(\mathcal{A})$ is polynomial, $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}})$ is polynomially bounded as desired. Since this implies a PPT adversary for the MSIS problem with non-negligible advantage, we arrive at a contradiction. This establishes that for any PPT $\mathcal{A}$, its advantage ϵ must be negligible.

To complete the proof of the main theorem, it remains to prove the following Lemmata 3.6 to 3.9.

Lemma 3.6. *We have* $\Pr[\mathsf{E}_3] \geq \frac{1}{2}\cdot \Pr[\mathsf{E}_2] - \mathsf{negl}(\lambda)$.

Proof. To analyze the success probability of using $\mathsf{Multi\text{-}Extract}$, we first construct a PPT adversary $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ against the straight-line extractability game (see Definition 2.10). On input $(1^\lambda, \widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}})$, $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ runs the Game_3 challenger $\mathcal{C}_3$ and simulates the view of Game_3 to $\mathcal{A}$. When $\mathcal{A}$ makes a random oracle query to $\mathsf{H_m}$, $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ relays it to the oracle $\mathcal{O}$ provided by the straight-line extractability game. It simulates the other oracle queries on-the-fly as $\mathcal{C}_3$. $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ also prepares a list L initially set to $\emptyset$, and when $\mathcal{A}$ submits the j-th ($j \in [\mathsf{Q_S}]$) first message $\rho_{j,1} = (\mathsf{com}_j, \pi^{\mathsf{m}}_j)$, $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ updates the list $L \leftarrow L \cup (\mathsf{X}_j = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}_j), \pi^{\mathsf{m}}_j)$. If $\mathcal{A}$ submits a valid forgery for the one-more unforgeability game, $\mathcal{B}_{\mathsf{Multi\text{-}Ext}}$ submits L as the $[\mathsf{Q_S}]$ statement and proof pairs.

Since $\mathcal{A}$ succeeds with probability $\Pr[\mathsf{E}_2]$, we have

$$\Pr\left[\{L = (\mathsf{X}_j, \pi_j)\}_{j\in[\mathsf{Q_S}]} \xleftarrow{\$} \mathcal{B}^{\mathcal{O}}_{\mathsf{Multi\text{-}Ext}}(1^\lambda, \widetilde{\mathsf{crs}}) : \forall j \in [\mathsf{Q_S}], \mathsf{Verify}^{\mathcal{O}}(\widetilde{\mathsf{crs}}, \mathsf{X}_j, \pi^{\mathsf{m}}_j) = \top\right] \geq \Pr[\mathsf{E}_2].$$

Then, by Definition 2.10, the probability that $\mathcal{A}$ outputs a valid forgery and $\mathsf{Multi\text{-}Extract}$ extracts a witness W_j such that $(\mathsf{X}_j, \mathsf{W}_j) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ for all $j \in [\mathsf{Q_S}]$ is at least $\frac{\Pr[\mathsf{E}_2]}{2} - \mathsf{negl}(\lambda)$. This establishes that $\mathsf{Multi\text{-}Extract}$ extracts all the witnesses if the challenger runs $\mathsf{Multi\text{-}Extract}$ *at the end* of the game. It remains to check that the challenger can run $\mathsf{Multi\text{-}Extract}$ during the game to arrive at the description of $\mathcal{C}_3$ in Game_3.

By noticing that the output of $\mathsf{Multi\text{-}Extract}$ is not used anywhere in Game_3, it is clear that the timing on which the challenger $\mathcal{C}_3$ runs $\mathsf{Multi\text{-}Extract}$ has no effect on the computed probability. It can run it during the game, rather than at the end of the game. Moreover, the challenger $\mathcal{C}_3$ may abort as soon as $\mathsf{Multi\text{-}Extract}$ fails to extract a witness without altering the success probability of $\mathcal{A}$ in Game_3. This completes the proof. □

Lemma 3.7. *We have* $\Pr[\mathsf{E}_4] \geq \frac{1}{\mathsf{Q_{H_M}}}\cdot\left(\Pr[\mathsf{E}_3] - \frac{\mathsf{Q}^2_{\mathsf{H_M}}+1}{|S_{\mathsf{hash}}|}\right)$.

Proof. Let us analyze $\Pr[\mathsf{Abort}_{\mathsf{guess}}]$. Firstly, we can assume every message in $\{\mathsf{M}_i\}_{i\in[\mathsf{Q_S}+1]}$ was queried to $\mathsf{H_M}$. This is because the probability that $\mathcal{A}$ can create a valid signature for these messages without querying $\mathsf{H_M}$ is at most $\frac{1}{|S_{\mathsf{hash}}|}$. Moreover, with all but probability $\frac{\mathsf{Q}^2_{\mathsf{H_M}}}{|S_{\mathsf{hash}}|}$, we can assume no hash collision is found. Conditioning on every messages in $\{\mathsf{M}_i\}_{i\in[\mathsf{Q_S}+1]}$ being queried to $\mathsf{H_M}$, we have $h_{j^*} \notin \{h'_j/c'_j\}_{j\in[\mathsf{Q_S}]}$ and $\mathsf{M}'_{j^*} \in \{\mathsf{M}_i\}_{i\in[\mathsf{Q_S}+1]}$ with probability at least $\frac{1}{\mathsf{Q_{H_M}}}$. This is because j^* is uniform random from the view of $\mathcal{A}$ and we may have $\mathsf{H_M}(\mathsf{M}_i) = h'_j/c'_j$ for $\mathsf{Q_S}$-pairs of $(i,j) \in [\mathsf{Q_S}+1]\times[\mathsf{Q_S}]$ in the worst case. In other words, there must exist at least one $\mathsf{M}'_{j^*} = \mathsf{M}_{i^*} \in \{\mathsf{M}_i\}_{i\in[\mathsf{Q_S}+1]}$ such that $\mathsf{H_M}(\mathsf{M}_{i^*}) \notin \{h'_j/c'_j\}_{j\in[\mathsf{Q_S}]}$ assuming there is no hash collision. Since the only differences between Game_3 and Game_4 are the abort condition, the statement follows. □

Lemma 3.8. *We have* $\Pr[\mathsf{E}_6] \geq \Pr[\mathsf{E}_5] - \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}}(\mathcal{B}_{\mathsf{MLWE}}) - \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}'_{\mathsf{DSMR}}) - 2 \cdot \mathsf{Adv}^{\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}_{\mathsf{DSMR}}) - \mathsf{negl}(\lambda)$*, where* $\mathcal{B}_{\mathsf{MLWE}}$*,* $\mathcal{B}'_{\mathsf{DSMR}}$*, and* $\mathcal{B}_{\mathsf{DSMR}}$ *are adversaries against the* $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$*,* $\mathsf{DSMR}_{d,k_1,\chi_{\mathsf{DSMR}},q,1}$*, and* $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ *problems, respectively, with* $\mathsf{Time}(\mathcal{B}_{\mathsf{MLWE}})$*,* $\mathsf{Time}(\mathcal{B}'_{\mathsf{DSMR}})$*, and* $\mathsf{Time}(\mathcal{B}_{\mathsf{DSMR}})$ *being roughly* $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$*.*

Proof. The proof consists of several hybrid games defined as follows, where we define $\mathsf{Game}_{5\text{-}1} := \mathsf{Game}_5$. and $\mathsf{Game}_{5\text{-}7} := \mathsf{Game}_6$.

$\mathsf{Game}_{5\text{-}2}$: The challenger modifies $\mathsf{crs}_{\mathsf{com}}$ and embeds a trapdoor in $(\mathbf{b}_i)_{i\in[k_2]} \subseteq \mathsf{crs}_{\mathsf{com}}$. For simplicity and without loss of generality, we assume $\widehat{\mathbf{b}} = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \in R_q^{k_2k_3}$ includes exactly one identity element $1 \in R_q$ and $k_2k_3 - 1$ elements that are uniform random over R_q. Although in general, $\widehat{\mathbf{b}}$ may include more 0 and 1, and possibly contain duplicate entries, these have no effect on the concrete proof as long as the number of uniform random elements are larger than $k_2 - 1$, which is necessary for any commitment scheme satisfying the hiding property.

Concretely, in this game, the challenger runs $(\widehat{\mathbf{b}}, \mathbf{T}_{\widehat{\mathbf{b}}}) \xleftarrow{\$} \mathsf{TrapGen}(1^{k_2k_3d}, q)$ and sets $\mathsf{crs}_{\mathsf{com}}$ to include them.[12] The rest remains the same as in the previous game. Then, by Lemma 2.19, the two games remain indistinguishable assuming the $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ assumption. Specifically, there exists a PPT adversary $\mathcal{B}_{\mathsf{DSMR}_1}$ against the $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ problem such that

$$\Pr[\mathsf{E}_{5\text{-}2}] \geq \Pr[\mathsf{E}_{5\text{-}1}] - \mathsf{Adv}^{\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}_{\mathsf{DSMR}_1}),$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{DSMR}_1})$ is roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$.

$\mathsf{Game}_{5\text{-}3}$: In this game, the challenger uses the trapdoor $\mathbf{T}_{\widehat{\mathbf{b}}}$ rather than $\mathbf{T}_{\mathbf{a}_1}$ to sample the short vector $\mathbf{e}$ such that

$$\left[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \widehat{\mathbf{b}}\right] \cdot \mathbf{e}^\top = u,$$

when it runs $\mathcal{S}_2$. Due to Lemma 2.20 and our parameter selection, we have

$$\Pr[\mathsf{E}_{5\text{-}3}] \geq \Pr[\mathsf{E}_{5\text{-}2}] - \mathsf{negl}(\lambda).$$

$\mathsf{Game}_{5\text{-}4}$: In this game, the challenger modifies $\mathbf{a}_1$. Rather than generating $\mathbf{a}_1$ along with a trapdoor $\mathbf{T}_{\mathbf{a}_1}$ using $\mathsf{TrapGen}$, the challenger simply samples $\mathbf{a}_1 \xleftarrow{\$} R_q^{k_1}$. Due to Lemma 2.19, this modification is indistinguishable assuming the $\mathsf{DSMR}_{d,k_1-1,\chi_{\mathsf{DSMR}},q,1}$ assumption. Specifically, there exists a PPT adversary $\mathcal{B}'_{\mathsf{DSMR}}$ against the $\mathsf{DSMR}_{d,k_1-1,\chi_{\mathsf{DSMR}},q,1}$ problem such that

$$\Pr[\mathsf{E}_{5\text{-}4}] \geq \Pr[\mathsf{E}_{5\text{-}3}] - \mathsf{Adv}^{\mathsf{DSMR}_{d,k_1-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}'_{\mathsf{DSMR}}),$$

where $\mathsf{Time}(\mathcal{B}'_{\mathsf{DSMR}})$ is roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$.

$\mathsf{Game}_{5\text{-}5}$: In this game, the challenger modifies $\mathbf{a}_2$. Rather than computing $\mathbf{a}_2$ as $\widetilde{\mathbf{a}}_2 - h_{j^*} \cdot \mathbf{g}$ where $\widetilde{\mathbf{a}}_2 \xleftarrow{\$} R_q^{k_2}$, the challenger samples $\mathbf{R} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1 \times k_2}$ and uses $\widetilde{\mathbf{a}}_2 = \mathbf{a}_1\mathbf{R}$. Recall that the first entry of $\mathbf{a}_1$ is the identity $1 \in R_q$ due to Lemma 2.19. Therefore, we can simply go through k_2-games to move from $\mathsf{Game}_{5\text{-}4}$ to $\mathsf{Game}_{5\text{-}5}$, where each adjacent games are indistinguishable assuming the $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}}}$ assumption.[13] Thus, there exists a PPT adversary $\mathcal{B}_{\mathsf{MLWE}}$ against the $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$ problem such that

$$\Pr[\mathsf{E}_{5\text{-}5}] \geq \Pr[\mathsf{E}_{5\text{-}4}] - k_2 \cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}}(\mathcal{B}_{\mathsf{MLWE}}),$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{MLWE}})$ is roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$. Here, note that the modification in $\mathsf{Game}_{5\text{-}3}$ and $\mathsf{Game}_{5\text{-}4}$ was crucial to construct $\mathcal{B}_{\mathsf{MLWE}}$.

[12] This is where we implicitly use the fact that $\mathsf{crs}_{\mathsf{com}}$ directly includes $(\mathbf{b}_i)_{i\in[k_2]}$, rather than assuming some efficient function mapping $\mathsf{crs}_{\mathsf{com}}$ to $(\mathbf{b}_i)_{i\in[k_2]}$.

[13] We note the proof works regardless of $\mathbf{a}_1$ including an identity element as long as it contains one invertible element in R_q, which we can assume without loss of generality.

$\mathsf{Game}_{5\text{-}6}$: In this game, the challenger no longer uses $\mathbf{T}_{\widehat{\mathbf{b}}}$ to sample the short vector $\mathbf{e}$. The challenger runs $\mathcal{S}_2$ as defined in Game_6, where it runs $\mathsf{SampleRight}$ using the extracted witness W_j instead of $\mathsf{SampleLeft}$ as in the previous game. It can be checked that $s_1(c\mathbf{R}')^2 \le s_1(c)^2 \cdot s_1(\mathbf{R})^2 + s_1(\widehat{\mathbf{R}})^2 \le B_c \cdot B_{\mathsf{MLWE}}^2 + \delta^{\mathsf{gap}2}$, where we used the fact that $s_1(\mathbf{R}) \le B_{\mathsf{MLWE}}$ with overwhelming probability and $s_1(\widehat{\mathbf{R}})$ is bounded by δ^{gap} by definition of $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. Then, due to our parameter selection and Lemma 2.20, and conditioning on event $\mathsf{Abort}_{\mathsf{extract}}$ and $\mathsf{Abort}_{\mathsf{guess}}$ not occurring, the distribution of the sampled vector remains $\mathsf{negl}(\lambda)$-close to the previous game. Therefore, we have

$$\Pr[\mathsf{E}_{5\text{-}6}] \ge \Pr[\mathsf{E}_{5\text{-}5}] - \mathsf{negl}(\lambda).$$

□

$\mathsf{Game}_{5\text{-}7}$: We undo the change we made in $\mathsf{Game}_{5\text{-}2}$ and use a uniform random $\mathsf{crs}_{\mathsf{com}}$ (and $(\mathbf{b}_i)_{i\in[k_2]}$). This game is identical to Game_6. Following the same argument we made to move from $\mathsf{Game}_{5\text{-}1}$ to $\mathsf{Game}_{5\text{-}2}$, there exists a PPT adversary $\mathcal{B}_{\mathsf{DSMR}_2}$ against the $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ problem such that

$$\Pr[\mathsf{E}_{5\text{-}7}] \ge \Pr[\mathsf{E}_{5\text{-}6}] - \mathsf{Adv}^{\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}_{\mathsf{DSMR}_2}).$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{DSMR}_2})$ is roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6)$.

Collecting the bounds and recalling $\Pr[\mathsf{E}_5] = \Pr[\mathsf{E}_{5\text{-}1}]$ and $\Pr[\mathsf{E}_6] = \Pr[\mathsf{E}_{5\text{-}7}]$, we arrive at the bound in the statement.

Lemma 3.9. *We have* $\mathsf{Adv}^{\mathsf{MSIS}_{d,1,k_2+k_2k_3,B_{\mathsf{MSIS}},q}}(\mathcal{B}_{\mathsf{MSIS}}) \ge \frac{1}{2p(\lambda)\cdot Q^{\mathsf{c}}_{\mathsf{H}_{\mathsf{s}}}} \cdot \Pr[\mathsf{E}_6]^{c_1} - \mathsf{negl}(\lambda)$, *where* $\mathcal{B}_{\mathsf{MSIS}}$ *is an adversary against the* $\mathsf{MSIS}_{d,1,k_2+k_2k_3,B_{\mathsf{MSIS}},q}$ *problem with* $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \le c_2 \cdot (\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6))$. *Here,* $p(\lambda)$ *is a polynomial, and* e, c_1, *and* c_2 *are constants defined in Definition 2.9.*

Proof. Before providing the description of $\mathcal{B}_{\mathsf{MSIS}}$, we first construct an adversary $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ against the single-proof extractability game (see Definition 2.9). Looking ahead, $\mathcal{B}_{\mathsf{MSIS}}$ runs $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ and $\mathsf{Single\text{-}Extract}^{\mathcal{B}_{\mathsf{Single\text{-}Ext}}}$ in order to (roughly) extract a solution to the MSIS problem.

Consider the statement of the form $\mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h)$, where $\mathbf{a}_1 \xleftarrow{\$} R_q^{k_1}$, $\mathbf{b}_i \xleftarrow{\$} R_q^{k_3}$ for $i \in [k_2]$, $h \xleftarrow{\$} S_{\mathsf{hash}}$, and $\mathbf{a}_2$ and u are further set as in Game_6. Denote this distribution as D_{X}. Then, on input X, $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ runs the Game_6 challenger $\mathcal{C}_6$ and simulates the view of Game_6 to $\mathcal{A}$, where $\mathcal{C}_6$ uses the contents in X to run the game. In particular, $\mathcal{C}_6$ uses $h \in \mathsf{X}$ instead of sampling h_{j^*} on its own. If any of the two events $\mathsf{Abort}_{\mathsf{extract}}$ and $\mathsf{Abort}_{\mathsf{guess}}$ occurred at some point during in the game, $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ outputs $\perp$. Otherwise, when $\mathcal{A}$ outputs its forgery $\{(\mathsf{M}_i, \Sigma_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$, if the event $\mathsf{Abort}_{\mathsf{guess}}$ did not occur, then we have $\mathsf{M}'_{j^*} \in \{\mathsf{M}_i\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$, where recall M'_{j^*} is the j^*-th ($j^* \in [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$) random oracle query to H_{M}. Let us denote $i^* \in [\mathsf{Q}_{\mathsf{S}}+1]$ as the unique $\mathsf{M}_{i^*} = \mathsf{M}'_{j^*}$ such that $\mathsf{H}_{\mathsf{M}}(\mathsf{M}_{i^*}) = h = h_{j^*}$, and parse $\pi^{\mathsf{s}} \leftarrow \Sigma_{i^*}$. Then, $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ outputs $\mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h)$ and $\pi = \pi^{\mathsf{s}}$.

It is easy to verify that $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ simulates the view to $\mathcal{A}$ perfectly, and we have

$$\Pr[\mathsf{E}_6] = \mathop{\mathbb{E}}_{\mathsf{X}\xleftarrow{\$} D_{\mathsf{X}}} \left[\Pr[(\mathsf{X},\pi) \xleftarrow{\$} \mathcal{B}^{\mathsf{H}_{\mathsf{s}}}_{\mathsf{Single\text{-}Ext}}(\mathsf{X}) : \mathsf{Verify}^{\mathsf{H}_{\mathsf{s}}}(\mathsf{X},\pi) = \top]\right], \tag{8}$$

where the probability is taken over the randomness used by $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$. Here, note that $\mathcal{B}_{\mathsf{Single\text{-}Ext}}$ uses the provided random oracle H_{s} to simulate oracle queries to H_{s} from $\mathcal{A}$, and simulates the rest of the random oracle queries on-the-fly using its randomness.

We are now ready to describe $\mathcal{B}_{\mathsf{MSIS}}$. Given an MSIS instance $\mathbf{d} = [\mathbf{a}_1 \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] = [\mathbf{a}_1 \mid \widehat{\mathbf{b}}] \in R_q^{k_2+k_2k_3}$, $\mathcal{B}_{\mathsf{MSIS}}$ prepares $\mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, u, h)$, where $\mathbf{a}_2$, u, and h are sampled as described above. It then executes $(\mathsf{X}, \mathsf{W}) \leftarrow \mathsf{Single\text{-}Extract}^{\mathcal{B}_{\mathsf{Single\text{-}Ext}}}(\mathsf{X})$. If $(\mathsf{X}, \mathsf{W}) \notin \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$, then $\mathcal{B}_{\mathsf{MSIS}}$ outputs $\perp$. Otherwise, it parses $((\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) := \tilde{\mathbf{e}}, c) \leftarrow \mathsf{W}$, where we have the following due to the definition of $\mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$:

$$\forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i} \ \wedge \ \|c\|_1 \le B_c \ \wedge \ \left[\mathbf{a}_1 \mid \mathbf{a}_2 + h\cdot\mathbf{g} \mid \widehat{\mathbf{b}}\right]\tilde{\mathbf{e}}^\top = c\cdot u. \tag{9}$$

Plugging in $\mathbf{a}_2 = \mathbf{a}_1\mathbf{R} - h\cdot\mathbf{g}$ and recalling u was generated in BSGen as $[\mathbf{a}_1 \mid \widehat{\mathbf{b}}]\cdot\mathbf{s}^\top$ for $\mathbf{s} \xleftarrow{\$} [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]_{\mathsf{coeff}}^{(k_1+k_2k_3)}$, the right hand equation can be rewritten as

$$\left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right]\begin{bmatrix} \tilde{\mathbf{e}}_1^\top + \mathbf{R}\tilde{\mathbf{e}}_2^\top \\ \tilde{\mathbf{e}}_3^\top \end{bmatrix} = c\cdot\left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right]\begin{bmatrix} \mathbf{s}_1^\top \\ \mathbf{s}_3^\top \end{bmatrix},$$

where $(\mathbf{s}_1, \mathbf{s}_3) := \mathbf{s} \in R^{k_1+k_2k_3}$. By subtracting both sides, we have

$$\left[\mathbf{a}_1 \mid \widehat{\mathbf{b}}\right]\underbrace{\begin{bmatrix} (\tilde{\mathbf{e}}_1^\top + \mathbf{R}\tilde{\mathbf{e}}_2^\top) - c\cdot\mathbf{s}_1^\top \\ \tilde{\mathbf{e}}_3^\top - c\cdot\mathbf{s}_3^\top \end{bmatrix}}_{=:\mathbf{z}^*} = \mathbf{0}.$$

$\mathcal{B}_{\mathsf{MSIS}}$ finally outputs $\mathbf{z}^* \in R^{k_1+k_2k_3}$ as a solution to the $\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}}}$ problem. Notice that

$$\begin{aligned}
\|\mathbf{z}^*\|_2 \leq& \|\tilde{\mathbf{e}}_1\|_2 + \|\tilde{\mathbf{e}}_3\|_2 + \|\mathbf{R}\tilde{\mathbf{e}}_2^\top\|_2 + \|c\cdot\mathbf{s}_1\|_2 + \|c\cdot\mathbf{s}_3\|_2 \\
\leq& B_{\Sigma,1}^{\mathcal{U},\mathsf{gap}} + B_{\Sigma,3}^{\mathcal{U},\mathsf{gap}} + s_1(\mathbf{R})B_{\Sigma,2}^{\mathcal{U},\mathsf{gap}} + \|c\|_1\cdot\|[\mathbf{s}_1 \mid \mathbf{s}_3]\|_2 \\
\leq& B_{\Sigma,1}^{\mathcal{U},\mathsf{gap}} + B_{\Sigma,3}^{\mathcal{U},\mathsf{gap}} + B_{\mathsf{MLWE}}B_{\Sigma,2}^{\mathcal{U},\mathsf{gap}} + B_c\cdot\Delta_{\mathsf{MLWE}}\sqrt{k_1+k_2k_3} = B_{\mathsf{MSIS}},
\end{aligned}$$

where the first inequality follows from the triangular inequality, the second inequality follows from the bounds $\|a\cdot b\|_2 \leq \|a\|_1\cdot\|b\|_2$, and $\|\mathbf{Ma}\|_2 \leq s_1(\mathbf{M})\cdot\|\mathbf{a}\|_2$.

It remains to analyze that $\mathbf{z}^*$ is a valid MSIS solution and that $\mathcal{B}_{\mathsf{MSIS}}$ outputs such $\mathbf{z}^*$ with non-negligible probability in polynomial time. For a fixed statement X, let us denote

$$\mu(\mathsf{X}) = \Pr[(\mathsf{X},\pi) \xleftarrow{\$} \mathcal{B}_{\mathsf{Single\text{-}Ext}}^{\mathsf{H}_\mathsf{s}}(\mathsf{X}) : \mathsf{Verify}^{\mathsf{H}_\mathsf{s}}(\mathsf{X},\pi) = \top].$$

By Eq. (8), we have $\Pr[\mathsf{E}_6] = \mathbb{E}_{\mathsf{X}\xleftarrow{\$}D_\mathsf{X}}[\mu(\mathsf{X})]$. Moreover, we have

$$\begin{aligned}
&\frac{1}{p(\lambda)\cdot\mathsf{Q}_{\mathsf{H}_\mathsf{s}}^e}\cdot\left(\mathop{\mathbb{E}}_{\mathsf{X}\xleftarrow{\$}D_\mathsf{X}}[\mu(\mathsf{X})]\right)^{c_1} - \mathsf{negl}(\lambda) \\
\leq&\frac{1}{p(\lambda)\cdot\mathsf{Q}_{\mathsf{H}_\mathsf{s}}^e}\cdot\mathop{\mathbb{E}}_{\mathsf{X}\xleftarrow{\$}D_\mathsf{X}}[\mu(\mathsf{X})^{c_1}] - \mathsf{negl}(\lambda) \\
\leq&\mathop{\mathbb{E}}_{\mathsf{X}\xleftarrow{\$}D_\mathsf{X}}\left[\Pr\left[\ \mathsf{W} \xleftarrow{\$} \mathsf{Single\text{-}Extract}^{\mathcal{B}_{\mathsf{Single\text{-}Ext}}}(\mathsf{X}) \ \ :\ \ (\mathsf{X},\mathsf{W}) \in \mathcal{R}_{\mathsf{gap}}^{\mathsf{s}}\ \right]\right] - \mathsf{negl}(\lambda) \\
=&\Pr[\mathbf{z}^* \xleftarrow{\$} \mathcal{B}_{\mathsf{MSIS}}(\mathbf{d}) : \|\mathbf{z}^*\|_2 \leq B_{\mathsf{MSIS}} \wedge \mathbf{d}\cdot\mathbf{z}^{*\top} = 0] - \mathsf{negl}(\lambda) \\
\leq&2\cdot\Pr[\mathbf{z}^* \xleftarrow{\$} \mathcal{B}_{\mathsf{MSIS}}(\mathbf{d}) : 0 < \|\mathbf{z}^*\|_2 \leq B_{\mathsf{MSIS}} \wedge \mathbf{d}\cdot\mathbf{z}^{*\top} = 0] - \mathsf{negl}(\lambda),
\end{aligned}$$

where the first inequality is due to Jensen's inequality, the second inequality is due to Definition 2.9, the third follows from the definition of $\mathcal{B}_{\mathsf{MSIS}}$ and the bound we established on $\mathbf{z}^*$, and the last inequality follows from the fact that there exists at least two distinct $\mathbf{s}, \mathbf{s}' \in [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]_{\mathsf{coeff}}^{(k_1+k_2k_3)}$ such that $u = [\mathbf{a}_1 \mid \mathbf{a}_2 \mid \widehat{\mathbf{b}}]\cdot\mathbf{s}^\top = [\mathbf{a}_1 \mid \mathbf{a}_2 \mid \widehat{\mathbf{b}}]\cdot\mathbf{s}'^\top$ due to our parameter selection. Specifically, from the view of $\mathcal{A}$ (and $\mathcal{C}_6$ by further noticing that $\mathcal{B}_{\mathsf{MSIS}}$ generated u), $\mathbf{s}$ has at least 1-bit of min-entropy, and thus, we have $\mathbf{z}^* \neq \mathbf{0}$ with probability at least $1/2$. This establishes that $\mathcal{B}_{\mathsf{MSIS}}$ outputs a non-zero $\mathbf{z}^*$ with the desired probability in the statement. We further have $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \leq c_2\cdot(\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_6))$ for some constant c_2 from Definition 2.9.

□

□

3.6 Extension: Partial Blind Signatures

We are able to obtain a *partially* blind signature [AO00] with a simple modification to our blind signature without increasing the signature size. Partially blind signatures are an extension of blind signatures where the message can contain a common message. This can be a message agreed between the user and the signer before the execution or a message that the signer would like to include for better system design, e.g., add an expiration date to revoke old signatures.

Our modification is simple. To bind the signature to a specific common message γ, the signer shifts the public syndrome $u \in R_q$ to $u - \mathsf{H}_{\mathsf{Mc}}(\gamma)$, where H_{Mc} is a newly introduced hash function that is modeled as a random oracle in the security proof. Since the construction and proof are almost identically, we refer the interested readers to Appendix C for the full details.

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 Section 3.2. We respectively provide in Sections 4.1 to 4.3, our concrete choices for the underlying 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 $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$. Finally, in Section 4.5, we explain the details on how to set the parameters for each building blocks and provide a concrete set of parameters for our resulting blind signature scheme.

4.1 Concrete Choice for Trapdoor-Sampling-Compatible Commitments

We rely on (a slight variant of) the BDLOP commitment by Baum et al. [BDL+18]. Below, we use two different moduli q' and q, where looking ahead, q is the modulus that explicitly shows up in the blind signature construction in the previous section. Although we can chose $q' = q$, it is better to chose them differently since informally q' and q are used by the MSIS and MLWE problems, respectively, and we can obtain better parameters by tuning them independently.

Construction. The commitment scheme Π_{Com} has message space $\mathcal{M} = R_q^L$ and randomness space $\mathcal{R} = [-1,1]_{\mathsf{coeff}}^{k_3 \times L}$, where recall $[-1,1]_{\mathsf{coeff}} \subset R_q$ denotes the set of polynomials with $\{-1,0,1\}$-coefficients. We first explain how the $\mathsf{crs}_{\mathsf{com}}$ is viewed.

$\mathsf{crs}_{\mathsf{com}}$: We assume the common random string $\mathsf{crs}_{\mathsf{com}}$ is of the following 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 $(\mathbf{b}'_0, \mathbf{b}'_1) \xleftarrow{\$} R_{q'}^{k_3-1} \times R_q^{k_3-2}$. Although we assumed $\mathsf{crs}_{\mathsf{com}}$ was a random binary string of length ℓ_{com} in Section 3, we can assume $\mathsf{crs}_{\mathsf{com}}$ is structured as above without loss of generality.

$\mathsf{Com}(\mathsf{crs}_{\mathsf{com}}, \mathsf{M})$: On input $\mathsf{crs}_{\mathsf{com}} = (\mathbf{b}_0, \mathbf{b}_1) \in R_{q'}^{k_3} \times R_q^{k_3}$, and messages $\vec{\mathsf{M}} = (\mathsf{M}_1, \cdots, \mathsf{M}_L) \in R_q^L$, it samples $\mathbf{R} \xleftarrow{\$} [-1,1]_{\mathsf{coeff}}^{k_3 \times L}$ and outputs

$$\mathsf{com} := \left(\begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \mathbf{R} + \begin{bmatrix} \mathbf{0} \\ \mathsf{M}_1 \mid \cdots \mid \mathsf{M}_L \end{bmatrix} \begin{matrix} \bmod q' \\ \bmod q \end{matrix} \right) \in R_{q'}^L \times R_q^L.$$

Here, the randomness used by the algorithm is $\mathsf{rand} := \mathbf{R}$.

The commitment scheme satisfies the following. We note that we do not explicitly require binding since this is implicitly handled by the soundness of the NIZKs.

Lemma 4.1. *The commitment scheme Π_{Com} is quantumly hiding and (k_3, δ)-trapdoor-sampling-compatible.*

Proof. The hiding property is shown to hold under the $\mathsf{MLWE}_{d,2,k_3-2,S_3,\max(q,q')}$ assumption in [BDL+18]. It remains to check that Π_{Com} satisfies all the properties provide in Definition 3.1. $\mathsf{ParseCom}(\mathsf{com})$ simply outputs the bottom half of com, i.e., $\mathbf{t} = \mathbf{b}_1\mathbf{R} + [\mathsf{M}_1 \mid \cdots \mid \mathsf{M}_L] \in R_q^L$, and $\mathsf{ParseRand}(\mathsf{rand})$ outputs the columns of $\mathbf{R}$. Since $\mathbf{R} \xleftarrow{\$} [-1,1]_{\mathsf{coeff}}^{k_3 \times L}$, we have that $s_1(\mathbf{R}) \le \delta = \sqrt{k_3 L} \cdot d$. □

4.2 Concrete Choice for Single-Proof Extractable NIZK

The single-proof extractable NIZK is based on the basic Lyubashevsky's sigma protocol [Lyu09, Lyu12], where soundness is argued through rewinding (or the forking lemma [PS00, BN06] 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.

Construction. The prove and verify algorithms of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ for the relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ are provided in Figs. 2 and 3, respectively. For reference, we recall below the relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ where we additionally take into consideration the unbalanced size of $\tilde{\mathbf{e}}$.

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

$\Pi^{\mathsf{s}}_{\mathsf{NIZK}} : \mathsf{Prove}^{\mathsf{H}_{\mathsf{s}}}(\mathsf{X}, \mathsf{W})$		(Implicit Verifier): X
$\mathsf{X} := (\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1, u, h) \in R_q^{k_1} \times R_q^{k_2} \times R_q^{k_3} \times R_q \times R_q,$ $\mathsf{W} := \tilde{\mathbf{e}} = (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) \in R^{k_1} \times R^{k_2} \times R^{k_3}$ s.t $\forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U}}_{\Sigma,i} \wedge \ [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1]\, \tilde{\mathbf{e}}^\top = u$		
For $i \in [3] : \mathbf{y}_i \xleftarrow{\$} D^{k_i}_{\gamma_{\mathbf{y}_i}}$		
$w := [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \begin{bmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \mathbf{y}_3^\top \end{bmatrix}$	$\boldsymbol{\alpha} := w \dashrightarrow$	
	$\dashleftarrow c$	$c := \mathsf{H}_{\mathsf{s}}(\mathsf{X}, \boldsymbol{\alpha}) \in S_{\mathsf{chal}} \subset R_q$
For $i \in [3] : \mathbf{z}_i := c \cdot \tilde{\mathbf{e}}_i + \mathbf{y}_i$		
If $\mathsf{Rej}(\mathbf{z}_1, c \cdot \tilde{\mathbf{e}}_1, \phi, T_1, \mathsf{err}) = \perp$ $\vee\ \mathsf{Rej}(\mathbf{z}_2, c \cdot \tilde{\mathbf{e}}_2, \phi, T_2, \mathsf{err}) = \perp$ $\vee\ \mathsf{Rej}(\mathbf{z}_3, c \cdot \tilde{\mathbf{e}}_3, \phi, T_3, \mathsf{err}) = \perp$	$\pi^{\mathsf{s}} \longrightarrow$	
then restart		
$\pi^{\mathsf{s}} := (c, (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3))$		

Figure 2: Prove algorithm for the single-proof NIZK for the relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$. For better readability, we illustrate the interactive protocol that underlies the NIZK. The dotted lines are internal to the prover, where it simulates the verifier of the interactive protocol (denoted as *implicit* verifier) using the hash function H_{s}. The solid line is the concrete output.

Security. The correctness of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ can be verified through a routine check. Below, we prove that $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ is classically zero-knowledge and single-proof extractable. The proof for the quantum setting is provided in Section 5.2.

Zero-Knowledge.

Theorem 4.2. *The* NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ *in Figs. 2 and 3 is classically zero-knowledge.*

Proof Sketch. The proof follows from those of the multi-proof NIZK, which we show later, since the single-proof NIZK is a major simplification of it. Moreover, the theorem is subsumed by prior results that show quantum zero-knowledge (or the stronger notion of simulation soundness) of the NIZK based on Lyubashevsky's sigma protocol, e.g., [KLS18, Kat21]. Here, unlike Theorem 4.4, we have *statistical* zero-knowledge since the underlying sigma protocol satisfies *statistical* honest-verifier zero-knowledge. □

Single-Proof Extractability.

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

$\mathsf{X} := (\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1, u, h) \in R_q^{k_1} \times R_q^{k_2} \times R_q^{k_3} \times R_q \times R_q,$
$\pi^{\mathsf{s}} := (c, (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)) \in R \times R^{k_1} \times R^{k_2} \times R^{k_3}$

$$w := [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix} - c \cdot u$$

$$\textbf{If} \begin{cases} \|\mathbf{z}_1\|_2 \geq B_{\Sigma,1} \\ \vee \ \|\mathbf{z}_2\|_2 \geq B_{\Sigma,2} \\ \vee \ \|\mathbf{z}_3\|_2 \geq B_{\Sigma,3} \\ \vee \ c \neq \mathsf{H}_\mathsf{s}(\mathsf{X}, w) \end{cases} \textbf{then return } \bot$$

return $\top$

Figure 3: Verify algorithm for the simplified single-proof NIZK for the relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$.

Theorem 4.3. *The* NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ *in Figs. 2 and 3 is classically single-proof extractable with* $(c_1, c_2, e) = (2, 2, 1)$ *and* $p(\lambda) = 1$.

Proof. By assumption, we have a PPT adversary $\mathcal{A}$ that makes at most $\mathsf{Q}_{\mathsf{H}_\mathsf{s}}$ random oracle queries such that for any $\mathsf{X} \in \mathcal{L}_{\mathcal{R}^{\mathsf{s}}}$,

$$\Pr[\pi^{\mathsf{s}} \xleftarrow{\$} \mathcal{A}^{\mathsf{H}_\mathsf{s}}(1^\lambda, \mathsf{X}) : \mathsf{Verify}^{\mathsf{H}_\mathsf{s}}(\mathsf{X}, \pi^{\mathsf{s}}) = \top] \geq \mu(\lambda). \tag{10}$$

Before describing the single-proof extractor Single-Extract, let us construct an adversary $\mathcal{B}$ against the forking lemma, i.e., Lemma A.1. $\mathcal{B}$ is given as input $\mathsf{par} = \mathsf{X}$ and $(c_1, \cdots, c_{\mathsf{Q}_{\mathsf{H}_\mathsf{s}}}) \xleftarrow{\$} S_{\mathsf{chal}}$, and internally runs $\mathcal{A}^{\mathsf{H}_\mathsf{s}}(1^\lambda, \mathsf{X})$, where it uses c_i to answer the i-th random oracle query made by $\mathcal{A}$. When $\mathcal{A}$ outputs π^{s}, $\mathcal{B}$ checks if π^{s} is valid and if $c \in (c_i)_{i \in [\mathsf{Q}_{\mathsf{H}_\mathsf{s}}]}$, where $(c, (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)) := \pi^{\mathsf{s}}$. If not, it outputs $(0, \sigma = \bot)$. Otherwise, it retrieves the smallest $J \in [\mathsf{Q}_{\mathsf{H}_\mathsf{s}}]$ such that $c = c_J$ and outputs $(J, \sigma = (w, c_J, (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)))$, where $w = [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \cdot [\mathbf{z}_1 | \mathbf{z}_2 | \mathbf{z}_3]^\top - c \cdot u$. By Eq. (10), we have

$$\mathsf{acc} = \Pr[(c_1, \cdots, c_{\mathsf{Q}_\mathsf{H}}) \xleftarrow{\$} S_{\mathsf{chal}}, (J, \sigma) \xleftarrow{\$} \mathcal{B}(\mathsf{X}, c_1, \cdots, c_{\mathsf{Q}_\mathsf{H}}) : J \geq 1] \geq \mu(\lambda).$$

We are now ready to specify Single-Extract. Single-Extract on input $\mathsf{X} \in \mathcal{L}_{\mathcal{R}^{\mathsf{s}}}$, runs $(b, \sigma_1, \sigma_2) \xleftarrow{\$} \mathsf{Fork}_{\mathcal{B}}(\mathsf{X})$ from Lemma A.1, which it can do with only black-box access to $\mathcal{A}$. If $b = 0$, then output $\bot$. Otherwise, if $(b, \sigma_1, \sigma_2) = (1, (w, c_I, (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)), (w', c'_I, (\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3)))$ for some $I \in [\mathsf{Q}_{\mathsf{H}_\mathsf{s}}]$, it outputs W, where $\mathsf{W} = (c_I - c_{I'}, \mathbf{z}_1 - \mathbf{z}'_1, \mathbf{z}_2 - \mathbf{z}'_2, \mathbf{z}_3 - \mathbf{z}'_3)$. We first check that W for $b = 1$ is a valid witness. If $\mathcal{A}$ outputs a forgery with respect to the I-th random oracle query, then by definition of $\mathsf{Fork}_{\mathcal{B}}$, the input w and w' to the random oracle must be the same since all the random oracle queries are answered identically in both runs up to the I-query. Since both proofs are valid, we have

$$[\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \cdot [\mathbf{z}_1 | \mathbf{z}_2 | \mathbf{z}_3]^\top - c_I \cdot u = [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \cdot [\mathbf{z}'_1 | \mathbf{z}'_2 | \mathbf{z}'_3]^\top - c_{I'} \cdot u.$$

Specifically, we have

$$[\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}_1] \cdot [\mathbf{z}_1 - \mathbf{z}'_1 | \mathbf{z}_2 - \mathbf{z}'_2 | \mathbf{z}_3 - \mathbf{z}'_3]^\top = (c_I - c_{I'}) \cdot u.$$

Hence, $\mathsf{W} = (c_I - c_{I'}, \mathbf{z}_1 - \mathbf{z}'_1, \mathbf{z}_2 - \mathbf{z}'_2, \mathbf{z}_3 - \mathbf{z}'_3)$ satisfies $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}}$, where we can set the bounds for each elements appropriately.

Finally, by Lemma A.1, the probability that Single-Extract outputs $\mathsf{W} \neq \bot$ is $\mathsf{frk} \geq \mathsf{acc} \cdot (\frac{\mathsf{acc}}{\mathsf{Q}_{\mathsf{H}_\mathsf{s}}} - \frac{1}{|S_{\mathsf{chal}}|}) \geq \frac{\mu(\lambda)^2}{\mathsf{Q}_{\mathsf{H}_\mathsf{s}}} - \mathsf{negl}(\lambda)$, if S_{chal} is exponentially large. Hence, $(c_1, c_2, e) = (2, 2, 1)$ and $p(\lambda) = 1$ in Definition 2.9 as desired. □

4.3 Concrete Choice for Multi-Proof Extractable NIZK

The statement we want to handle is of the form $\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\end{bmatrix}\mathbf{R} + \begin{bmatrix}0\\ h\cdot\mathbf{g}\end{bmatrix}$ for private $\mathbf{R}$ and h. As explained in the technical overview, we first (implicitly) construct a single-proof rewinding NIZK by combination of the exact proof of Bootle et al. [BLS19] and a proof for linear relations. This allows us to prove *exact* soundness of $\mathbf{R}$ and *relaxed* soundness of h. We note that it is unclear how to prove exact soundness of h using Bootle et al.

We then rely on the Katsumata transform [Kat21] to add multi-proof straight-extractability to this base single-proof rewinding protocol. This transform uses an *extractable linear homomorphic commitment*, which is in other words, a linear homomorphic PKE with pseudo-random public keys. At a high level, the idea is to modify the prover to further encrypt/commit the witness $\mathsf{W} = (\mathbf{R}, h)$ and randomness rand used by the underlying base protocol. Then, during in the security proof, the reduction generates the public key with an associated decryption key and tries to decrypt the ciphertext $\mathsf{ct}_{\mathsf{W}}, \mathsf{ct}_{\mathsf{rand}}$. Unfortunately, this simple reduction fails since the prover never proves that the ciphertexts ct_{W} and $\mathsf{ct}_{\mathsf{rand}}$ really encrypt the witness and randomness during the real protocol. That is, there is no guarantee that the cheating prover encrypted something useful. However, the prover does prove that the added ciphertext $c\cdot\mathsf{ct}_{\mathsf{W}} + \mathsf{ct}_{\mathsf{rand}}$ is well formed, where c is a challenge output by the random oracle. Informally, this means that given, possibly maliciously generated, ciphertexts ct_{W} and $\mathsf{ct}_{\mathsf{rand}}$, there should intuitively exist a non-negligible fraction of challenges c for which $c\cdot\mathsf{ct}_{\mathsf{W}} + \mathsf{ct}_{\mathsf{rand}}$ is a valid ciphertext. Therefore, at a high level, the reduction samples many challenges c and attempts to decrypt $c\cdot\mathsf{ct}_{\mathsf{W}} + \mathsf{ct}_{\mathsf{rand}}$ rather than trying to individually decrypt ct_{W} and $\mathsf{ct}_{\mathsf{rand}}$. When decryption succeeds several times, it can rely on the underlying base protocols special soundness to extract a witness.

Although the intuition is clear, turning this idea into a formal proof requires a careful probability analysis on the adversary's success probability. This section includes the most non-trivial proof techniques and we believe it has independent interest.

Preparation. Let us prepare some notations. Let $R_{q'} = \mathbb{Z}_{q'}/(X^d+1)$ 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:

- $\mathcal{R}^{\mathsf{m}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com}), \\ \mathsf{W} = (h, \mathsf{rand} := \mathbf{R}) \end{array} \middle| \begin{array}{c} h \in S_{\mathsf{hash}} \wedge \mathbf{R} \in [-1,1]^{k_3\times L}_{\mathsf{coeff}}, \\ \wedge\ \mathsf{com} = \left(\begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\end{bmatrix}\mathbf{R} + \begin{bmatrix}\mathbf{0}\\ h\cdot\mathbf{g}\end{bmatrix} \begin{array}{l}\mod q'\\ \mod q\end{array}\right) \end{array} \right\};$
- $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}} := \left\{ \begin{array}{c} \mathsf{X} = (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com}), \\ \mathsf{W} = (h', c, (\mathbf{r}_i)_{i\in[L]}) \end{array} \middle| \begin{array}{c} \|h'\|_2 \le B_{\mathsf{inv}}/2 \ \wedge\ \|c\|_1 \le B_c \ \wedge\ \mathbf{t} = \mathsf{ParseCom}(\mathsf{com}) \\ \wedge\ \mathbf{R} \in [-1,1]^{k_3\times L}_{\mathsf{coeff}} \ \wedge\ \forall i \in [L], t_i = \mathbf{b}_1\mathbf{r}_i^\top + (h'/c)\cdot q^{\frac{i-1}{L}} \end{array} \right\},$

where recall $\mathbf{g} = [1 \mid q^{\frac{1}{L}} \mid \cdots \mid q^{\frac{L-1}{L}}] \in R_q^L$ is the gadget matrix. Notice the gap relation $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ has no slack for the commitment randomness. It can be checked that we recover $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ in Section 3.2 by setting $\delta^{\mathsf{gap}} = \sqrt{k_3 L}\cdot d$. That is, any $\mathbf{R}$ with $\{-1,0,1\}$-coefficient polynomial entries has spectral norm smaller than δ^{gap}.

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. 4 and 5, respectively. For better readability of the proof and following prior conventions [BLS19, Kat21], 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: $R_{q'} = \mathbb{Z}_{q'}/(X^d+1)$ is a fully splitting ring that is used for Bootle et al's [BLS19] exact proof; $R_q = \mathbb{Z}_q/(X^d+1)$ is a ring where any small element is invertible and is used for the linear proof; $R_Q = \mathbb{Z}_Q/(X^d+1)$ is used for the the multi-proof straight-line extractability as in [Kat21], 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 Section 4.5.

Security. Keeping in mind that any $r \in R_{q'}$ with coefficients in $\{0,1,2\}$ satisfy $\Phi(r) \circ (\Phi(r)-1) \circ (\Phi(r)-2) = \mathbf{0}$, the correctness of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ can be verified through a routine (but tedious) check. Below, we prove that $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ is classically zero-knowledge and multi-proof extractable. The proof for the quantum setting is provided in Section 5.3.

Zero-Knowledge.

Theorem 4.4. *The* NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ *in Figs. 4 and 5 is classically zero-knowledge if the* $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{d}'},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}'},Q}$, *and* $\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}$ *problems are hard.*

Proof. The proof consists of two parts. In the first part, we show that the *interactive* protocol that underlies our NIZK is honest-verifier zero-knowledge. That is, if we use an honest verifier instead of the hash function H_{m} to generate the challenges, then it is zero-knowledge. We then show that if the underlying interactive protocol is (non-abort) honest-verifier zero-knowledge, then the resulting NIZK is zero-knowledge. More precisely, we first show the following.

Lemma 4.5. *Consider an interactive protocol as defined in Fig. 4 except that the verifier samples the challenges* $(\mathbf{c}_1, \mathbf{c}_2) \xleftarrow{\$} \mathbb{Z}^{\tau}_{q'} \times \left(C_X^{\tau\cdot\tau'} \times C_{\mathsf{ham}}\right)$ *and the prover responds with* a_1 *and* a_2 *in the first and third flows, respectively, and the following* resp *in the fifth flow:*

$$\mathsf{resp} := \left((\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{i,j \in [\tau] \times [\tau']}, \zeta, \mathbf{Z}', f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\right).$$

Here resp *is the same as* π^{m} *after removing the redundant elements included in* $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2)$. *Let* $D^{\not\perp}_{\mathsf{trans}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathsf{W})$ *be the distribution of a transcript* $\mathsf{trans} := (a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$ *from the honest interactive protocol with prover input* $(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathsf{W})$ *conditioned on not restarting/aborting. Then there exists a simulator* $\mathsf{Sim}_{\mathsf{int}}$ *such that for any* $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}^{\mathsf{m}}$ *and PPT (or possibly a QPT)* $\mathcal{A}$, *we have*

$$\left|\Pr\left[\begin{array}{c}(\mathbf{c}_1, \mathbf{c}_2) \xleftarrow{\$} \mathbb{Z}^{\tau}_{q'} \times \left(C_X^{\tau\cdot\tau'} \times C_{\mathsf{ham}}\right), \\ (a_1, a_2, \mathsf{resp}) \xleftarrow{\$} \mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}_1, \mathbf{c}_2)\end{array} : \mathcal{A}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, (a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})) = 1\right]\right.$$
$$\left. - \Pr\left[\mathsf{trans} \xleftarrow{\$} D^{\not\perp}_{\mathsf{trans}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathsf{W}) : \mathcal{A}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{trans}) = 1\right]\right| = \mathsf{negl}(\lambda),$$

where the probability is also taken over the randomness of sampling $\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}} = (H, \mathbf{a}_0, (\mathbf{A}_k)_{k \in [4]})$.

Proof. Below, we show in a sequence of games that the output of $\mathsf{Sim}_{\mathsf{int}}$ is indistinguishable from an honest non-aborting transcript. Observe the real prover algorithm in Fig. 4. At a high level, we modify this in three steps: we first simulate the text highlighted in gray corresponding to the straight-line extractability; we then simulate the texts in gray corresponding to the exact sound proof; finally, we simulate the texts in black without any highlights corresponding to the proof for linear relations.

Game_0 : In this game, the adversary $\mathcal{A}$ is given the real transcript trans. We denote by ϵ_0 the probability that $\mathcal{A}$ outputs 1.

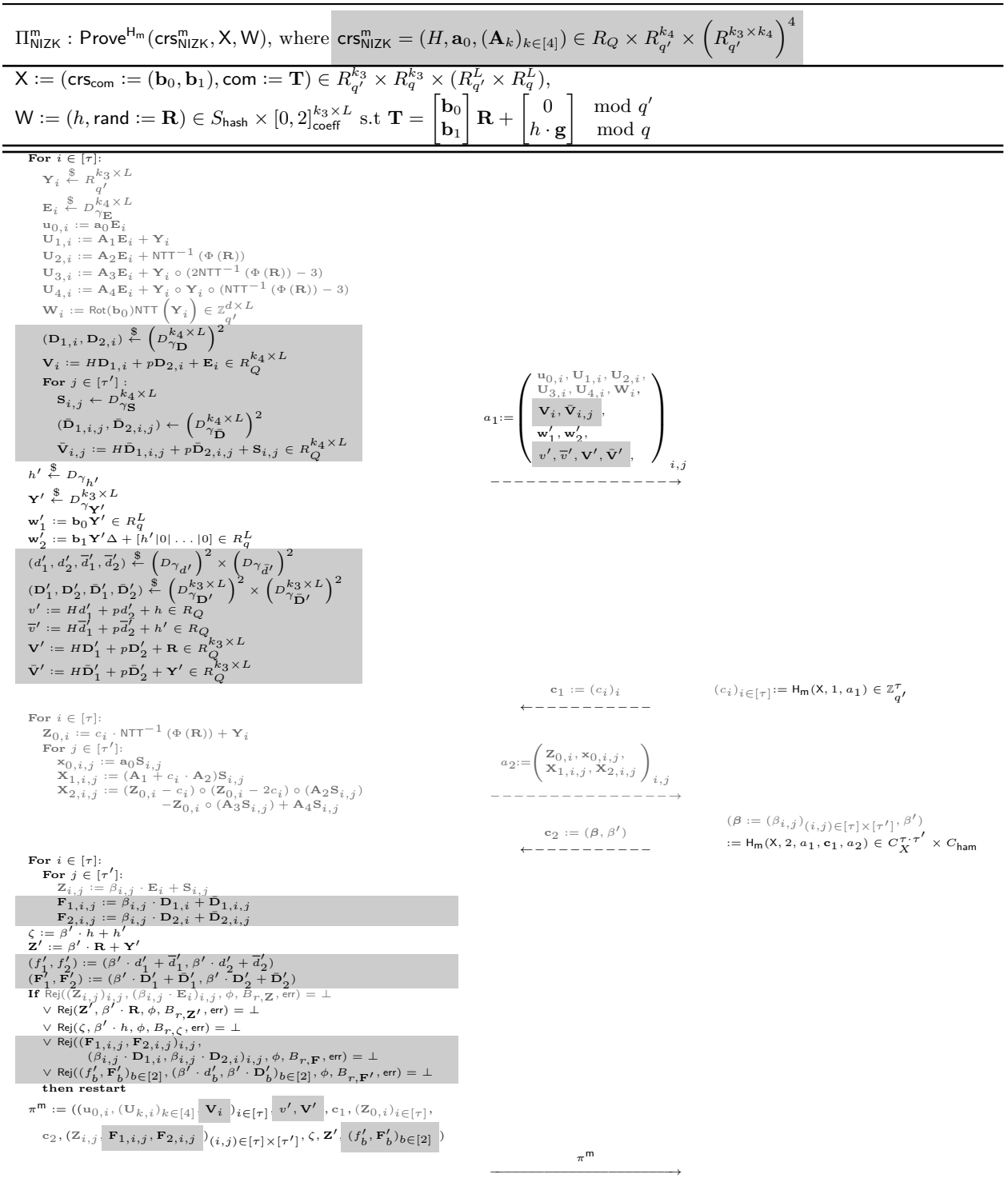

Figure 4: Prove algorithm for the multi-proof NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$. The crs for $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ consists of a random element H (used for extraction) and random matrices $(\mathbf{a}_0, (\mathbf{A}_k)_{k\in[4]})$ (used for committing), and the crs for Π_{Com} is a random tuple $(\mathbf{b}_0, \mathbf{b}_1)$. For better readability, we illustrate the 5-round interactive protocol that implicitly underlies the NIZK. The dotted lines are internal to the prover, where it simulates the verifier of the interactive protocol using the hash function $\mathsf{H_m}$. The solid line is the concrete output of the NIZK. The texts in gray are used by the exact proof of [BLS19], 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 [Kat21].

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

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

$\pi^{\mathsf{m}} := ((\mathbf{u}_{0,i}, (\mathbf{U}_{k,i})_{k\in[4]}, \mathbf{V}_i)_{i\in[\tau]}, v', \mathbf{V}', \mathbf{c}_1, (\mathbf{Z}_{0,i})_{i\in[\tau]}, \mathbf{c}_2, (\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, (f'_b, \mathbf{F}'_b)_{b\in[2]})$

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

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

return $\top$

Figure 5: Verify algorithm for the multi-proof NIZK for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$. The texts in gray are used by the exact proof of [BLS19], 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 [Kat21].

$\mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}_1, \mathbf{c}_2)$, where $\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}} = (H, \mathbf{a}_0, (\mathbf{A}_k)_{k\in[4]}) \in R_Q \times R_{q'}^{k_4} \times \left(R_{q'}^{k_3\times k_4}\right)^4$

$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T}) \in R_{q'}^{k_3+1} \times R_q^{k_3+1} \times (R_{q'}^L \times R_q^L)$

$\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix} := \mathbf{T} \in R_{q'}^L \times R_q^L$
$(c_i)_{i\in[\tau]} := \mathbf{c}_1$
$(\boldsymbol{\beta} = (\beta_{i,j})_{i,j\in[\tau]\times[\tau']}, \beta') := \mathbf{c}_2$
$(\zeta, \mathbf{Z}') \xleftarrow{\$} D_{\gamma_{h'}} \times D_{\gamma_{\mathbf{Y}'}}^{k_3\times L}$
$\mathbf{w}'_1 := \mathbf{b}_0\mathbf{Z}'^{\top} - \beta'\mathbf{t}_1$
$\mathbf{w}'_2 := \mathbf{b}_1\mathbf{Z}'^{\top}\Delta - \beta'\mathbf{t}_2\Delta - [\zeta\|0\|\dots\|0]$
$(f'_1, f'_2) \xleftarrow{\$} \left(D_{\gamma_{\bar{d}'}}\right)^2$
$(\mathbf{F}'_1, \mathbf{F}'_2) \xleftarrow{\$} \left(D_{\gamma_{\bar{\mathbf{D}}'}}^{k_3\times L}\right)^2$
$(v', \mathbf{V}') \xleftarrow{\$} R_Q \times R_Q^{k_3\times L}$
$\overline{v}' := Hf'_1 + pf'_2 + \zeta - \beta'\cdot v'$
$\bar{\mathbf{V}}' := H\mathbf{F}'_1 + p\mathbf{F}'_2 + \mathbf{Z}' - \beta'\cdot\mathbf{V}'$
For $i \in [\tau]$:
  $\mathbf{Z}_{0,i} \xleftarrow{\$} \mathbb{Z}_{q'}^{k_3\times L}$
  $\mathbf{u}_{0,i} \xleftarrow{\$} R_Q^L$
  $(\mathbf{U}_{k,i})_{k\in[4]} \xleftarrow{\$} \left(R_Q^{k_4\times L}\right)^4$
  $\mathbf{V}_i \xleftarrow{\$} R_Q^{k_4\times L}$
  $\mathbf{W}_i := \mathsf{Rot}(\mathbf{b}_0)\mathsf{NTT}\left(\mathbf{Z}_{0,i}\right) - c_i\cdot\Phi\left(\mathbf{t}_1\right) \in \mathbb{Z}_{q'}^{d\times L}$
  For $j \in [\tau']$:
    $\mathbf{Z}_{i,j} \xleftarrow{\$} D_{\gamma_{\mathbf{S}}}^{k_4\times L}$
    $(\mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j}) \xleftarrow{\$} \left(D_{\gamma_{\bar{\mathbf{D}}}}^{k_4\times L}\right)^2$
    $\bar{\mathbf{V}}_{i,j} := H\mathbf{F}_{1,i,j} + p\mathbf{F}_{2,i,j} + \mathbf{Z}_{i,j} - \beta_{i,j}\cdot\mathbf{V}_i$
    $\mathbf{x}_{0,i,j} := \mathbf{a}_0\mathbf{Z}_{i,j} - \beta_{i,j}\cdot\mathbf{u}_{0,i}$
    $\mathbf{X}_{1,i,j} := (\mathbf{A}_1 + c_i\cdot\mathbf{A}_2)\mathbf{Z}_{i,j} + \beta_{i,j}\cdot\left(\mathbf{Z}_{0,i} - (\mathbf{U}_{1,i} + c_i\cdot\mathbf{U}_{2,i})\right)$
    $\mathbf{X}_{2,i,j} := (\mathbf{Z}_{0,i} - c_i)\circ(\mathbf{Z}_{0,i} - 2c_i)\circ(\mathbf{A}_2\mathbf{Z}_{i,j}) - \mathbf{Z}_{0,i}\circ(\mathbf{A}_3\mathbf{Z}_{i,j}) + \mathbf{A}_4\mathbf{Z}_{i,j}$
      $-\beta_{i,j}\cdot\left((\mathbf{Z}_{0,i} - c_i)\circ(\mathbf{Z}_{0,i} - 2c_i)\circ\mathbf{U}_{2,i} - \mathbf{Z}_{0,i}\circ\mathbf{U}_{3,i} + \mathbf{U}_{4,i}\right)$
$a_1 := \left(\left(\mathbf{u}_{0,i}, \mathbf{U}_{1,i}, \mathbf{U}_{2,i}, \mathbf{U}_{3,i}, \mathbf{U}_{4,i}, \mathbf{W}_i, \mathbf{V}_i, (\bar{\mathbf{V}}_{i,j})_{j\in[\tau']}\right)_{i\in[\tau]}, \mathbf{w}'_1, \mathbf{w}'_2, v', \overline{v}', \mathbf{V}', \bar{\mathbf{V}}'\right)$
$a_2 := \left(\mathbf{Z}_{0,i}, \mathbf{x}_{0,i,j}, (\mathbf{X}_{1,i,j}, \mathbf{X}_{2,i,j})_{j\in[\tau']}\right)_{i\in[\tau]}$
$\mathsf{resp} := ((\mathbf{Z}_{i,j}\ \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j}\ )_{(i,j)\in[\tau]\times[\tau']}, \zeta, \mathbf{Z}', f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\ )$
return $(a_1, a_2, \mathsf{resp})$

Figure 6: Simulator for the interactive protocol underlining the multi-proof NIZK for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$. The texts in gray are used by the exact proof of [BLS19], 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 [Kat21].

Game_1 : In this game, we modify the texts highlighted in gray in Fig. 4. We sample $(\mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{i,j\in[\tau]\times[\tau']} \xleftarrow{\$} \left(D_{\gamma_{\bar{\mathbf{D}}}}^{k_4\times L}\right)^2$ for $(i,j) \in [\tau]\times[\tau']$, $(f'_1, f'_2) \xleftarrow{\$} \left(D_{\gamma_{\bar{d}'}}\right)^2$, and $(\mathbf{F}'_1, \mathbf{F}'_2) \xleftarrow{\$} \left(D_{\gamma_{\bar{\mathbf{D}}'}}^{k_3\times L}\right)^2$. We then set $\left(\overline{v}', \overline{\mathbf{V}}', (\overline{\mathbf{V}}_{i,j})_{(i,j)\in[\tau]\times[\tau']}\right)$ as in Fig. 6. Namely, we set these terms in reverse order while maintaining consistency of the verification algorithm in Fig. 5. Due to a standard argument using the rejection sampling (cf. Lemma 2.12), we have

$$|\epsilon_0 - \epsilon_1| \leq \mathsf{negl}(\lambda).$$

Game_2 : In this game, we further modify the texts highlighted in gray in Fig. 4. We sample $\left(v', \mathbf{V}', (\mathbf{V}_i)_{i\in[\tau]}\right)$ uniformly random over their respective domains instead of setting them as MLWE instances. Since $\left((\mathbf{D}_{1,i}, \mathbf{D}_{2,i})_{i\in[\tau]}, d'_1, d'_2, \mathbf{D}'_1, \mathbf{D}'_2\right)$ are now distributed independently from $\left((\mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\right)$ due to the modification we made in Game_1, we can construct PPT adversaries $\mathcal{B}_{\mathsf{MLWE},1}, \mathcal{B}_{\mathsf{MLWE},2}$, and $\mathcal{B}_{\mathsf{MLWE},3}$ against the $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{d}'},Q}$, and $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}'},Q}$ problems, respectively, such that

$$\begin{aligned}|\epsilon_1 - \epsilon_2| \leq\ & 2\cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,1,1,\gamma_{\bar{d}'},Q}}(\mathcal{B}_{\mathsf{MLWE},1}) \\ & + L\cdot k_3 \cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}'},Q}}(\mathcal{B}_{\mathsf{MLWE},2}) \\ & + \tau\cdot L\cdot k_4 \cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}}(\mathcal{B}_{\mathsf{MLWE},3}).\end{aligned}$$

Game_3 : In this game, we modify part of the text in gray in Fig. 4. We sample $\mathbf{Z}_{0,i} \xleftarrow{\$} R_{q'}^{k_3\times L}$ and $\mathbf{Z}_{i,j} \xleftarrow{\$} D_{\gamma_{\mathbf{S}}}^{k_4\times L}$ for $(i,j) \in [\tau]\times[\tau']$. We then set $\left(\mathbf{W}_i, \mathbf{x}_{0,i,j}, (\mathbf{X}_{1,i,j}, \mathbf{X}_{2,i,j})_{j\in[\tau']}\right)_{i\in[\tau]}$ as in Fig. 6. Namely, we set these terms in reverse order while maintaining consistency of the verification algorithm in Fig. 5. Due to the modification we made in the previous game and by a standard argument using the rejection sampling (cf. Lemma 2.12), we have

$$|\epsilon_2 - \epsilon_3| \leq \mathsf{negl}(\lambda).$$

Game_4 : In this game, we change the rest of the text in gray as in Fig. 6. Specifically, the only difference between the prior game is that $(\mathbf{u}_{0,i}, (\mathbf{U}_{k,i})_{k\in[4]})$ are now sampled uniformly random, rather than being generated as an MLWE instance. Since $(\mathbf{E}_i)_{i\in[\tau]}$ are distributed independently from $\left(\mathbf{V}_i, (\overline{\mathbf{V}}_{i,j}\mathbf{Z}_{i,j})_{j\in[\tau']}\right)_{i\in[\tau]}$ due to the modifications we made in Game_2 and Game_3, we can construct a PPT adversary $\mathcal{B}_{\mathsf{MLWE},4}$ against the $\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}$ problem, such that

$$|\epsilon_3 - \epsilon_4| \leq \tau\cdot L\cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}}(\mathcal{B}_{\mathsf{MLWE},4}).$$

Game_5 : In this game, we modify the texts in black without the gray highlights in Fig. 4. We sample $(\zeta, \mathbf{Z}') \xleftarrow{\$} D_{\gamma_{h'}} \times D_{\gamma_{\mathbf{Y}'}}^{k_3\times L}$ and then set $(\mathbf{w}'_1, \mathbf{w}'_2)$ as is Fig. 6. Due to the modification we made in Game_2 and by a standard argument using the rejection sampling (cf. Lemma 2.12), we have

$$|\epsilon_4 - \epsilon_5| \leq \mathsf{negl}(\lambda).$$

Notice that the simulator in Game_5 is identical to $\mathsf{Sim}_{\mathsf{int}}$ provided in Fig. 6. Moreover, the proof remains exactly the same even against QPT adversaries. Thus, this completes the proof of Lemma 4.5. □

To finish the proof of Theorem 4.4, we provide in Fig. 7 the zero-knowledge simulator $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ for $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ that internally runs $\mathsf{Sim}_{\mathsf{int}}$. Recall Sim_0 simulates the hash function H_{m}, and Sim_1 simulates the NIZK proof. At a high level, we use the simulator $\mathsf{Sim}_{\mathsf{int}}$ of the underlying interactive protocol to simulate the proof. For every statement $\mathsf{X} \in \mathcal{L}_{\mathcal{R}}$, we sample a random challenge $(\mathbf{c}_1, \mathbf{c}_2)$ and then execute $(a_1, a_2, \mathsf{resp}) \xleftarrow{\$} \mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}_1, \mathbf{c}_2)$. We then "patch" the random oracle (simulated by Sim_0) so it outputs $\mathbf{c}_1$ and $\mathbf{c}_2$ on input $(\mathsf{X}, 1, a_1)$ and $(\mathsf{X}, 2, a_1, \mathbf{c}_1, a_2)$, respectively.

More formally, in Fig. 7, we introduce an algorithm $\mathsf{GetTrans}$, which internally runs $\mathsf{Sim}_{\mathsf{int}}$. Recalling that the adversary can obtain at most two proofs per statement (see Definition 2.8), the bit b taken as input to $\mathsf{GetTrans}$ controls which proof to provide. The randomness used to run $\mathsf{Sim}_{\mathsf{int}}$ is generated by a $\mathsf{Q}_{\mathsf{H\text{-}P}}$-wise independent hash function $\widehat{\mathsf{H}}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}$, where $\mathsf{Q}_{\mathsf{H\text{-}P}}$ is the total number of queries the adversary makes to H_{m} and Prove. Here, note that using $\widehat{\mathsf{H}}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}$ remains perfectly indistinguishable from a random function from an adversary making at most $\mathsf{Q}_{\mathsf{H\text{-}P}}$ queries.[14] Sim_0 and Sim_1 are defined in the natural way using $\mathsf{GetTrans}$, where Sim_0 uses an additional $\mathsf{Q}_{\mathsf{H\text{-}P}}$-wise independent hash function $\mathsf{H}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}$ to output random elements when it is not queried on the patched point.[15]

The only difference between having oracle access to $(\mathsf{Sim}_0, \mathcal{S})$ and $(\mathsf{H}_{\mathsf{m}}, \mathsf{Prove})$ is whether the proof is generated honestly or by $\mathsf{Sim}_{\mathsf{int}}$. Hence, since the view of any PPT adversary $\mathcal{A}$ is altered (either implicitly or explicitly) by at most $\mathsf{Q}_{\mathsf{H\text{-}P}}$ proofs, we have $\mathsf{Adv}^{ZK}_{\Pi_{\mathsf{NIZK}}}(\mathcal{A}) = \mathsf{negl}(\lambda)$ due to Lemma 4.5. Here, note that $\mathsf{Sim}_{\mathsf{int}}$ is only indistinguishable from a "non-aborting" honest prover Prove. However, since the probability of Prove outputting $\perp$ is negligible, this only negligibly alters the adversary's advantage.

This completes the proof of Theorem 4.4. □

$\mathsf{Sim}_0(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathsf{flow}, \mathbf{a})$

$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T})$

For $b \in \{0,1\}$
  $(\mathbf{c}^*_{b,1}, \mathbf{c}^*_{b,2}, \mathsf{rand}_{b,\mathsf{int}}) \leftarrow \widehat{\mathsf{H}}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}(\mathsf{X}, b)$
  $(a^*_{b,1}, a^*_{b,2}, \mathsf{resp}^*_b) \leftarrow \mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}^*_{b,1}, \mathbf{c}^*_{b,2}; \mathsf{rand}_{b,\mathsf{int}})$
If $\mathsf{flow} = 1$ **then**
  If $\mathbf{a} = a^*_{0,1}$ **then return** $\mathbf{c}^*_{0,1}$
  ElseIf $\mathbf{a} = a^*_{1,1}$ **then return** $\mathbf{c}^*_{1,1}$
If $\mathsf{flow} = 2$ **then**
  If $\mathbf{a} = (a^*_{0,1}, \mathbf{c}^*_{0,1}, a^*_{0,2})$ **then return** $\mathbf{c}^*_{0,2}$
  ElseIf $\mathbf{a} = (a^*_{1,1}, \mathbf{c}^*_{1,1}, a^*_{1,2})$ **then return** $\mathbf{c}^*_{1,2}$
Else return $\mathsf{H}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}(\mathsf{X}, \mathsf{flow}, a)$

$\mathsf{Sim}_1(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}; L)$

$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T})$

If $L[\mathsf{X}] = \phi$ **then**
  $L[\mathsf{X}] := \top$
  return $\mathsf{GetTrans}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, 0)$
Else
  return $\mathsf{GetTrans}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, 1)$

$\mathsf{GetTrans}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, b)$

$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T})$

$(\mathbf{c}_{b,1}, \mathbf{c}_{b,2}, \mathsf{rand}_{b,\mathsf{int}}) \leftarrow \widehat{\mathsf{H}}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}(\mathsf{X}, b)$
$(a^*_{b,1}, a^*_{b,2}, \mathsf{resp}^*_b) \leftarrow \mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}^*_{b,1}, \mathbf{c}^*_{b,2}; \mathsf{rand}_{\mathsf{int}})$
$\left(\left(\mathbf{u}_{0,i}, \mathbf{U}_{1,i}, \mathbf{U}_{2,i}, \mathbf{U}_{3,i}, \mathbf{U}_{4,i}, \mathbf{W}_i, \mathbf{V}_i, (\bar{\mathbf{V}}_{i,j})_{j\in[\tau']}\right)_{i\in[\tau]}, \mathbf{w}'_1, \mathbf{w}'_2, v', \overline{v}', \mathbf{V}', \bar{\mathbf{V}}'\right) := a_{b,1}$
$\left(\mathbf{Z}_{0,i}, \mathbf{x}_{0,i,j}, (\mathbf{X}_{1,i,j}, \mathbf{X}_{2,i,j})_{j\in[\tau']}\right)_{i\in[\tau]} := a_{b,2}$
$((\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \zeta, \mathbf{Z}', f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2) := \mathsf{resp}_b$
return $\pi^{\mathsf{m}} := ((\mathbf{u}_{0,i}, (\mathbf{U}_{k,i})_{k\in[4]}, \mathbf{V}_i)_{i\in[\tau]}, v', \mathbf{V}', \mathbf{c}_{b,1}, (\mathbf{Z}_{0,i})_{i\in[\tau]},$
$\mathbf{c}_{b,2}, (\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2)$

Figure 7: Zero-knowledge simulator $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ of the multi-proof NIZK for the relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$. $\mathsf{H}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}$ and $\widehat{\mathsf{H}}_{\mathsf{m}}^{\mathsf{Q}_{\mathsf{H\text{-}P}}\text{-wise}}$ are two $\mathsf{Q}_{\mathsf{H\text{-}P}}$-wise independent hash functions: the former maps to $\mathbb{Z}_{q'}^{\tau}$; the latter maps to $C_X^{\tau\cdot\tau'} \times C_{\mathsf{ham}}$ and the randomness space used by simulator $\mathsf{Sim}_{\mathsf{int}}$ for the simplified interactive protocol. List L, initially set to empty, stores the number of time the adversary queries X to Sim_1, which is at most 2 by Definition 2.8.

Multi-Proof Extractability.

[14] We can rely on standard *lazy-sampling* strategies where the simulator samples the output of the random oracle only when it is queried. We rely on a proof strategy that fixes the description of the ROM once and for all to make the proof consistent with those in the QROM.

[15] Formally, we would need to define two $\mathsf{Q}_{\mathsf{H\text{-}P}}$-wise independent hash functions since the output space of $\mathsf{H}_{\mathsf{m}}(\mathsf{X}, 1, a_1)$ and $\mathsf{H}_{\mathsf{m}}(\mathsf{X}, 2, a_1, \mathbf{c}_1, a_2)$ are different. We omit this subtlety without loss of generality to maintain readability.

Theorem 4.6. *The* NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ *in Figs. 4 and 5 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. The simulator $\mathcal{S}_{\mathsf{crs}}$ samples $(f, v) \xleftarrow{\$} \chi^2_{\mathsf{DSMR}}$ and $(\mathbf{a}_0, (\mathbf{A}_k)_{k\in[4]}) \xleftarrow{\$} R_Q \times R_{q'}^{k_4} \times \left(R_{q'}^{k_3\times k_4}\right)^4$ conditioned on f being invertible over R_Q, and then outputs $\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}} = (H, \mathbf{a}_0, (\mathbf{A}_k)_{k\in[4]})$ and $\tau = (f, v)$, where $H = p \cdot v \cdot f^{-1} \in R_Q$. This is indistinguishable from the random distribution of the real $\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}$ based on the $\mathsf{DSMR}_{d,1,\chi_{\mathsf{DSMR}},Q,p}$ assumption.

Straight-Line Extractability. The proof consists of three parts. We first show in Lemma 4.7 that (roughly) if the adversary $\mathcal{A}$ outputs a valid proof, then $\mathcal{A}$ must have been able 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 4.10 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 $\mathsf{Multi\text{-}Extract}$ 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 4.7. *Consider an interactive protocol as defined implicitly in Fig. 4. 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* $i \in [\tau]$*:*

$$\Gamma_{1,i}(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2) := \left\{ \overline{c}_i \in \mathbb{Z}_{q'} \;\middle|\; \begin{array}{l} (c_{i'})_{i'\in[\tau]} \leftarrow \mathbf{c}_1, \overline{\mathbf{c}}_1 := (\overline{c}_i) \cup (c_{i'})_{i'\in[\tau]\setminus\{i\}}, \\ (\boldsymbol{\beta} = (\beta_{i',j'})_{(i',j')\in[\tau]\times[\tau']}, \beta') \leftarrow \mathbf{c}_2 \\ \exists j \in [\tau'], \textit{distinct } (\overline{\beta}_{i,j}, \overline{\beta}'_{i,j}) \in (C_X)^2, \\ \overline{\boldsymbol{\beta}} := (\overline{\beta}_{i,j}) \cup \left(\beta_{i',j'}\right)_{(i',j')\neq(i,j)}, \overline{\boldsymbol{\beta}}' := (\overline{\beta}'_{i,j}) \cup \left(\beta_{i',j'}\right)_{(i',j')\neq(i,j)}, \\ \exists(\overline{a}_2, \overline{a}'_2), (\overline{\mathsf{resp}}, \overline{\mathsf{resp}}') \textit{ s.t. } (a_1, \overline{\mathbf{c}}_1, \overline{a}_2, \mathbf{c}_2 := (\overline{\boldsymbol{\beta}}, \beta'), \overline{\mathsf{resp}}) \textit{ and} \\ (a_1, \overline{\mathbf{c}}_1, \overline{a}'_2, \overline{\mathbf{c}}_2 := (\overline{\boldsymbol{\beta}}', \beta'), \overline{\mathsf{resp}}') \textit{ are valid} \end{array} \right\}$$

$$\Gamma_2(\mathsf{X}, a_1, \mathbf{c}_1, a_2, \mathbf{c}_2) := \left\{ \overline{\beta}' \in C_{\mathsf{ham}} \mid (\boldsymbol{\beta}, \beta') \leftarrow \mathbf{c}_2, \exists \overline{\mathsf{resp}} \textit{ s.t. } (a_1, \mathbf{c}_1, a_2, \mathbf{c}_2 := (\boldsymbol{\beta}, \overline{\beta}'), \overline{\mathsf{resp}}) \textit{ is valid} \right\},$$

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,

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

Proof. For notational simplicity, we denote $\Gamma_{1,i}^{(k)} := \Gamma_{1,i}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$ and $\Gamma_2^{(k)} := \Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$ for each $(k,i) \in [\mathsf{Q}_\mathsf{S}] \times [\tau]$. Let T_2 be a positive integer, which we define shortly after. We denote by $\mathsf{ValidProofs}$ the event that $\mathsf{Verify}^{\mathsf{H}_\mathsf{m}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}_k, \pi_k^{\mathsf{m}}) = \top$ for all $k \in [\mathsf{Q}_\mathsf{S}]$ and, when the context is clear, we omit the sampling probability space. Then, we can rewrite $\mathcal{A}$'s advantage as follows:

$$
\begin{aligned}
\mu &\le \Pr[\mathsf{ValidProofs}] \\
&= \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_\mathsf{S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_2^{(k)}\right| \ge T_2\right)\right)\right] \\
&\quad + \Pr\left[\mathsf{ValidProofs} \wedge \left(\exists k \in [\mathsf{Q}_\mathsf{S}], \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right) \vee \left(\left|\Gamma_2^{(k)}\right| < T_2\right)\right)\right] \\
&\le \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_\mathsf{S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_2^{(k)}\right| \ge T_2\right)\right)\right] \\
&\quad + \sum_{k \in [\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right] \qquad (11) \\
&\quad + \sum_{k \in [\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left|\Gamma_2^{(k)}\right| < T_2\right] \qquad (12) \\
&\le \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_\mathsf{S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_2^{(k)}\right| \ge T_2\right)\right)\right] \\
&\quad + \frac{\mathsf{Q}_\mathsf{H}^2}{2} \cdot \left(\frac{2}{q} + \frac{1}{(2d)^{\tau'}}\right)^{\tau} + \mathsf{Q}_\mathsf{H} \cdot \frac{T_2}{|C_{\mathsf{ham}}|},
\end{aligned}
$$

where the second inequality follows from the union bound, and the third inequality is due to Corollaries 4.8 and 4.9 that establish upper bounds on Eqs. (11) and (12), respectively. We first finish the proof of Lemma 4.7.

By plugging in $T_2 := \frac{\mu}{2\mathsf{Q}_\mathsf{H}} \cdot |C_{\mathsf{ham}}|$ in the above inequality, we obtain the following

$$
\Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_\mathsf{S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_2^{(k)}\right| \ge T_2\right)\right)\right] \ge \frac{\mu}{2} - \frac{\mathsf{Q}_\mathsf{H}^2}{2} \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau}.
$$

Due to our parameter setting (i.e., $\frac{q'}{2} \approx (2d)^{\tau'}$ and $1/(2d)^{\tau \cdot \tau'} = \mathsf{negl}(\lambda)$), for any $\mathsf{Q}_\mathsf{H} = \mathsf{poly}(\lambda)$, $\mathsf{Q}_\mathsf{H} \cdot \left(\frac{2}{q} + \frac{1}{(2d)^{\tau'}}\right)^{\tau}$ is negligible. Thus we obtain the desired bound.

It remains to prove the following Corollaries 4.8 and 4.9.

Corollary 4.8. *We have* $\sum_{k \in [\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right] \le \frac{\mathsf{Q}_\mathsf{H}^2}{2} \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau}$.

Proof. We further modify the equation as follows,

$$
\begin{aligned}
&\sum_{k \in [\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\right] \\
&= \sum_{k \in [\mathsf{Q}_\mathsf{S}]} \sum_{J_k \in [0:\tau]} \sum_{\substack{S \subseteq [\tau] \\ \text{s.t. } |S| = J_k}} \Pr\left[\begin{array}{c} \mathsf{ValidProofs} \\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right) \end{array} \wedge \begin{pmatrix} \forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)} \\ \forall i \in [\tau] \backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)} \end{pmatrix}\right] \\
&\le \sum_{k \in [\mathsf{Q}_\mathsf{S}]} \sum_{J_k \in [0:\tau]} \sum_{\substack{S \subseteq [\tau] \\ \text{s.t. } |S| = J_k}} \Pr\left[\begin{pmatrix} \forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)} \\ \forall i \in [\tau] \backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)} \end{pmatrix} \middle| \begin{array}{c} \mathsf{ValidProofs} \\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right) \end{array}\right] \\
&\le \sum_{k \in [\mathsf{Q}_\mathsf{S}]} \sum_{J_k \in [0:\tau]} \sum_{\substack{S \subseteq [\tau] \\ \text{s.t. } |S| = J_k}} \Pr\left[\left(\forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c} \mathsf{ValidProofs} \\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right) \end{array}\right]
\end{aligned}
$$

$$\cdot \Pr\left[\left(\forall i \in [\tau]\backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c}\mathsf{ValidProofs} \\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right] \tag{13}$$

where $c_{1,k,i}$ is the i-th element in the k-th second-flow challenge $\mathbf{c}_{1,k}$ included in π_k^{m} output by adversary . The first inequality follows from taking the conditional probability and the second inequality follows from the fact that the output of the random oracle is uniform and thus the distributions of each $(c_{1,k,i})_{i\in[\tau]}$ are independent (even though $\mathcal{A}$ can freely chose which $(c_{1,k,i})_{i\in[\tau]}$ to output). In other words, for each $k \in [\mathsf{Q}_\mathsf{S}]$ and $\mathbf{c}_{1,k} = (c_{1,k,i})_{i\in[\tau]}$, $c_{1,k,i}$ is either in $\Gamma_{1,i}^{(k)}$ of size at most 2 or not, and J_k counts the number of $c_{1,k,i} \in \Gamma_{1,i}^{(k)}$ in $\mathbf{c}_{i,k}$.

We first consider the probability that $c_{1,k,i} \in \Gamma_{1,i}^{(k)}$ for all $i \in S$. Let us fix $k \in [\mathsf{Q}_\mathsf{S}]$, $J_k \in [0:\tau]$, and $S \subseteq [\tau]$ such that $|S| = J_k$. Since $\mathbf{c}_{1,k} \in \mathbb{Z}_{q'}^{\tau}$ is only defined once after the adversary queries the random oracle on input $(\mathsf{X}_k, a_{1,k})$, $\{c_{1,k,i}\}_{i\in S}$ is distributed uniformly random over $\mathbb{Z}_{q'}^{J_k}$ before the adversary queries the random oracle. For simplicity, we assume without loss of generality that the adversary always queries $(\mathsf{X}_k, a_{1,k})$ to the random oracle before outputting its purported proofs and each statement X_k are different. Then, the probability that $c_{1,k,i} \in \Gamma_{1,i}^{(k)}$ for all $i \in S$ is $\mathsf{Q}_{\mathsf{H},k} \cdot \prod_{i\in S} \frac{\left|\Gamma_{1,i}^{(k)}\right|}{q'} \le \mathsf{Q}_{\mathsf{H},k} \cdot \left(\frac{2}{q'}\right)^{J_k}$, where $\mathsf{Q}_{\mathsf{H},k}$ denotes the number of random oracle queries that include $(\mathsf{X}_k, 1)$ as a prefix and we use the fact $\left|\Gamma_{1,i}^{(k)}\right| \le 2$.

We next consider the probability that $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\backslash S$. By definition of $\Gamma_{1,i}^{(k)}$, this is equivalent to the fact that there exists only a unique $(\beta_{k,i,j})_{j\in[\tau']}$ such that the transcript is valid. Similarly to above, $((\beta_{k,i,j})_{j\in[\tau']})_{i\in[\tau]\backslash S}$ is distributed uniformly random over $C_X^{\tau'(\tau-J_k)}$ before the adversary queries the random oracle. Then, the probability that $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\backslash S$ is $\overline{\mathsf{Q}}_{\mathsf{H},k} \cdot \left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k}$, where $\overline{\mathsf{Q}}_{\mathsf{H},k}$ denotes the number of random oracle queries that include $(\mathsf{X}_k, 2, a_{1,k}, \mathbf{c}_{1,k})$ as a prefix and we use the fact that $|C_X| = 2d$.

Combining the two arguments, we upper bound Eq. (13) as follow:

$$\begin{aligned}
&\sum_{k\in[\mathsf{Q}_\mathsf{S}]} \sum_{J_k\in[0:\tau]} \sum_{\substack{S\subseteq[\tau] \\ \text{s.t. } |S|=J_k}} \mathsf{Q}_{\mathsf{H},k} \cdot \left(\frac{2}{q'}\right)^{J_k} \cdot \overline{\mathsf{Q}}_{\mathsf{H},k} \cdot \left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k} \\
&= \sum_{k\in[\mathsf{Q}_\mathsf{S}]} \sum_{J_k\in[0:\tau]} \binom{\tau}{J_k} \mathsf{Q}_{\mathsf{H},k} \cdot \overline{\mathsf{Q}}_{\mathsf{H},k} \cdot \left(\frac{2}{q'}\right)^{J_k} \cdot \left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k} \\
&= \sum_{k\in[\mathsf{Q}_\mathsf{S}]} \mathsf{Q}_{\mathsf{H},k} \cdot \overline{\mathsf{Q}}_{\mathsf{H},k} \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau} \\
&\le \frac{\mathsf{Q}_\mathsf{H}^2}{2} \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau},
\end{aligned}$$

where the second equality follows from the binomial expansion and the last inequality follows from $\sum_{k\in[\mathsf{Q}_\mathsf{S}]}(\mathsf{Q}_{\mathsf{H},k} + \overline{\mathsf{Q}}_{\mathsf{H},k}) \le \mathsf{Q}_\mathsf{H}$. This completes the proof. □

Corollary 4.9. *We have* $\sum_{k\in[\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left|\Gamma_2^{(k)}\right| < T_2\right] \le \mathsf{Q}_\mathsf{H} \cdot \frac{T_2}{|C_{\mathsf{ham}}|}$.

Proof. Similarly to the proof of Corollary 4.8, β_k' is distributed uniformly random over C_{ham} before the adversary queries $(\mathsf{X}_k, 2, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k})$ to the random oracle. Therefore,

$$\sum_{k\in[\mathsf{Q}_\mathsf{S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left|\Gamma_2^{(k)}\right| < T_2\right] \le \sum_{k\in[\mathsf{Q}_\mathsf{S}]} \mathsf{Q}_{\mathsf{H},k} \cdot \frac{\left|\Gamma_2^{(k)}\right|}{|C_{\mathsf{ham}}|} \le \mathsf{Q}_\mathsf{H} \cdot \frac{T_2}{|C_{\mathsf{ham}}|},$$

where $\mathsf{Q}_{\mathsf{H},k}$ denotes the number of random oracle queries that include $(\mathsf{X}_k, 2, a_{1,k}, \mathbf{c}_{1,k})$ as a prefix. □

□

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 it. 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 beginning of this section, 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 4.10. *Consider the following γ valid transcripts for a statement X:*

- *For $(\eta, b) \in [3] \times [2]$,*
  $\mathsf{trans}^{(\eta,b)} := \big(a_1, \mathbf{c}_1^{(\eta)} := (c_i^{(\eta)})_{i\in[\tau]}, a_2^{(\eta)}, \mathbf{c}_2^{(\eta,b)} := (\boldsymbol{\beta}^{(\eta,b)} := (\beta_{i,j}^{(\eta,b)})_{(i,j)\in[\tau]\times[\tau']}, \beta'), \mathsf{resp}^{(\eta,b)}\big)$,
- $\widehat{\mathsf{trans}}^{(1,0)} := (a_1, \mathbf{c}_1^{(1)}, a_2^{(1)}, \widehat{\mathbf{c}}_2^{(1,b)} := (\boldsymbol{\beta}^{(1,0)}, \widehat{\beta}'), \widehat{\mathsf{resp}}^{(1,0)})$,

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 $\Big(\mathsf{trans}^{(1,0)}, \big(\beta_{i^*,j_\eta^*}^{(\eta,0)}, \beta_{i^*,j_\eta^*}^{(\eta,1)}\big)_{\eta\in[3]}, \big(\beta', \widehat{\beta}'\big)\Big)$ included in any of the γ 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. For reference, we recall what $\mathsf{trans}^{(1,0)}$ contains:

- $a_1 := \Big((\mathbf{u}_{0,i}, \mathbf{U}_{1,i}, \mathbf{U}_{2,i}, \mathbf{U}_{3,i}, \mathbf{U}_{4,i}, \mathbf{W}_i, \mathbf{V}_i, (\bar{\mathbf{V}}_{i,j})_{j\in[\tau']})_{i\in[\tau]}, \mathbf{w}_1', \mathbf{w}_2', v', \overline{v}', \mathbf{V}', \bar{\mathbf{V}}'\Big)$,
- $\mathbf{c}_1^{(1)} := (c_i^{(1)})_{i\in[\tau]} \in \mathbb{Z}_{q'}^{\tau}$,
- $a_2^{(1)} := \Big(\mathbf{Z}_{0,i}^{(1)}, (\mathbf{x}_{0,i,j}^{(1)}, \mathbf{X}_{1,i,j}^{(1)}, \mathbf{X}_{2,i,j}^{(1)})_{j\in[\tau']}\Big)_{i\in[\tau]}$,
- $\mathbf{c}_2^{(1,0)} := (\boldsymbol{\beta}^{(1,0)} := (\beta_{i,j}^{(1,0)})_{(i,j)\in[\tau]\times[\tau']}, \beta') \in C_X^{\tau\cdot\tau'} \times C_{\mathsf{ham}}$,
- $\mathsf{resp}^{(1,0)} := \big((\mathbf{Z}_{i,j}^{(1,0)}, \mathbf{F}_{1,i,j}^{(1,0)}, \mathbf{F}_{2,i,j}^{(1,0)})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}^{(1,0)'}, \zeta^{(1,0)}, f_1^{(1,0)'}, f_2^{(1,0)'}, \mathbf{F}_1^{(1,0)'}, \mathbf{F}_2^{(1,0)'}\big)$.

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 in Section 4.4 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:

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

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. 5). $\mathsf{Extract}_{\mathsf{ss}}$ subtracts these equations to remove $\mathbf{w}_1'$ and $\mathbf{w}_2'$, and obtains the following:

$$\begin{aligned}(\beta' - \widehat{\beta}') \cdot \mathbf{t}_1 &= \mathbf{b}_0\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big) \text{ (over } R_{q'}),\\ (\beta' - \widehat{\beta}') \cdot \mathbf{t}_2\Delta &= \mathbf{b}_1\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big)\Delta + [\zeta^{(1,0)} - \widehat{\zeta}^{(1,0)} \mid 0 \mid \cdots \mid 0] \text{ (over } R_q).\end{aligned} \tag{14}$$

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

$$(\beta' - \widehat{\beta}') \cdot \mathbf{t}_2 = \mathbf{b}_1\big(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\big) + (\zeta^{(1,0)} - \widehat{\zeta}^{(1,0)}) \cdot \mathbf{g}. \tag{15}$$

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. If this is not the case, $\mathsf{Extract}_{\mathsf{ss}}$ proceeds as follows. 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 will only output a solution to the MSIS problem in the following Parts (B) and (C).

Part (B). Following the argument from the previous part, $\mathsf{Extract}_{\mathsf{ss}}$ first performs the following NTRU decryption for $(\eta, b) \in [3] \times [2]$, which is guaranteed to succeed by assumption:

$$\mathbf{Z}^{(\eta,b)}_{i^*,j^*_\eta} := f^{-1} \cdot (f \cdot (\bar{\mathbf{V}}_{i^*,j^*_\eta} + \beta^{(\eta,b)}_{i^*,j^*_\eta} \cdot \mathbf{V}_{i^*}) \mod Q) \mod p.$$

Fix $\eta \in [3]$. Then, since $\mathsf{trans}^{(\eta,0)}$ and $\mathsf{trans}^{(\eta,1)}$ are valid transcripts, they satisfy the same verification equation regarding $\mathbf{x}^{(\eta)}_{0,i^*,j^*_\eta}$ (see Fig. 5). Subtracting both sides, $\mathsf{Extract}_{\mathsf{ss}}$ thus obtains the following for all $\eta \in [3]$:

$$\mathbf{a}_0 \underbrace{\big(\mathbf{Z}^{(\eta,0)}_{i^*,j^*_\eta} - \mathbf{Z}^{(\eta,1)}_{i^*,j^*_\eta}\big)}_{=:\widetilde{\mathbf{Z}}^{(\eta)}_{i^*,j^*_\eta}} = \underbrace{(\beta^{(\eta,0)}_{i^*,j^*_\eta} - \beta^{(\eta,1)}_{i^*,j^*_\eta})}_{=:\widetilde{\beta}^{(\eta)}_{i^*,j^*_\eta}} \cdot \mathbf{u}_{0,i^*} \text{ (over } R_{q'}).$$

By [BCK+14, Lemma 2.1], any difference of distinct challenges in C_X is small and invertible over $R_{q'}$. Here, we note that we do not care if the inverse is small. Thus, if there exists $\eta_1, \eta_2 \in [3]$ such that $\widetilde{\mathbf{Z}}^{(\eta_1)}_{i^*,j^*_{\eta_1}}/\widetilde{\beta}^{(\eta_1)}_{i^*,j^*_{\eta_1}} \neq \widetilde{\mathbf{Z}}^{(\eta_2)}_{i^*,j^*_{\eta_2}}/\widetilde{\beta}^{(\eta_2)}_{i^*,j^*_{\eta_2}}$, then $\mathsf{Extract}_{\mathsf{ss}}$ outputs

$$\mathbf{S} := \widetilde{\beta}^{(\eta_2)}_{i^*,j^*_{\eta_2}} \cdot \widetilde{\mathbf{Z}}^{(\eta_1)}_{i^*,j^*_{\eta_1}} - \widetilde{\beta}^{(\eta_1)}_{i^*,j^*_{\eta_1}} \cdot \widetilde{\mathbf{Z}}^{(\eta_2)}_{i^*,j^*_{\eta_2}}$$

as an $\mathsf{MSIS}_{d,1,k_4,16B_{\mathbf{Z}},q'}$ solution for $\mathbf{a}_0$. Otherwise, $\mathsf{Extract}_{\mathsf{ss}}$ computes $\mathsf{M}_{k,i^*} := \mathbf{U}_{k,i^*} - \mathbf{A}_k\widetilde{\mathbf{Z}}^{(\eta)}_{i^*,j^*_\eta}/\widetilde{\beta}^{(\eta)}_{i^*,j^*_\eta}$ over $R_{q'}$ for all $k \in [4]$, which is in particular independent from the choice of $\eta \in [3]$. Before finishing

explaining the description of $\mathsf{Extract}_{\mathsf{ss}}$, which we provide in Part (C), we make a detour and claim that $\mathbf{t}_1 = \mathbf{b}_0\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*}))$ and $\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*})) \in R^{k_3\times L}$ consists of $\{0,1,2\}$-coefficient polynomials.

Although $\mathsf{Extract}_{\mathsf{ss}}$ is not given $(c_{i^*}^{(\eta)}, \mathbf{Z}_{0,i^*}^{(\eta)}, \mathbf{X}_{1,i^*,j_\eta^*}^{(\eta)})_{\eta\in\{2,3\}}$ as input, it is guaranteed that such elements exist and satisfy the following verificaton equations regarding $\mathbf{X}_{1,i^*,j_\eta^*}^{(\eta)}$ for $(\eta, b) \in [3]\times[2]$ by assumption (see Fig. 5).

$$(\mathbf{A}_1 + c_{i^*}^{(\eta)}\cdot\mathbf{A}_2)\mathbf{Z}_{i^*,j_\eta^*}^{(\eta,b)} + \beta_{i^*,j_\eta^*}^{(\eta,b)}\cdot\mathbf{Z}_{0,i^*}^{(\eta)} = \mathbf{X}_{1,i^*,j_\eta^*}^{(\eta)} + \beta_{i^*,j_\eta^*}^{(\eta,b)}\cdot(\mathbf{U}_{1,i^*} + c_{i^*}^{(\eta)}\cdot\mathbf{U}_{2,i^*})$$

For each $\eta\in[3]$, we can subtract the equations for $b=1$ and 2 to remove (the unknown) $\mathbf{X}_{1,i^*,j_\eta^*}^{(\eta)}$ as follows:

$$(\mathbf{A}_1 + c_{i^*}^{(\eta)}\cdot\mathbf{A}_2)\widetilde{\mathbf{Z}}_{i^*,j_\eta^*}^{(\eta)} + \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot\mathbf{Z}_{0,i^*}^{(\eta)} = \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot(\mathbf{U}_{1,i^*} + c_{i^*}^{(\eta)}\cdot\mathbf{U}_{2,i^*})$$

Further substituting the commitment openings for $\mathbf{U}_{1,i^*}$ and $\mathbf{U}_{2,i^*}$ with the appropriate choice of $\eta\in[3]$, we obtain

$$(\mathbf{A}_1 + c_{i^*}^{(\eta)}\cdot\mathbf{A}_2)\widetilde{\mathbf{Z}}_{i^*,j_\eta^*}^{(\eta)} + \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot\mathbf{Z}_{0,i^*}^{(\eta)} = \mathbf{A}_1\widetilde{\mathbf{Z}}_{i^*,j_\eta^*} + \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot\mathsf{M}_{1,i^*,j_\eta^*} + c_{i^*}^{(\eta)}\cdot\left(\mathbf{A}_2\widetilde{\mathbf{Z}}_{i^*,j_\eta^*} + \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot\mathsf{M}_{2,i^*,j_\eta^*}\right).$$

Routine calculation shows that $\mathbf{Z}_{0,i^*}^{(\eta)} = \mathsf{M}_{1,i^*} + c_{i^*}^{(\eta)}\cdot\mathsf{M}_{2,i^*}$. Performing the exact same argument on the verification equations regarding $\mathbf{X}_{2,i^*,j_\eta^*}^{(\eta)}$ (see Fig. 5), we obtain the following for each $\eta\in[3]$.

$$\begin{aligned}&(\mathbf{Z}_{0,i^*}^{(\eta)} - c_{i^*}^{(\eta)})\circ(\mathbf{Z}_{0,i^*}^{(\eta)} - 2c_{i^*}^{(\eta)})\circ(\mathbf{A}_2\widetilde{\mathbf{Z}}_{i^*,j_\eta^*}^{(\eta)}) - \mathbf{Z}_{0,i^*}^{(\eta)}\circ(\mathbf{A}_3\widetilde{\mathbf{Z}}_{i^*,j_\eta^*}^{(\eta)}) + \mathbf{A}_4\widetilde{\mathbf{Z}}_{i^*,j_\eta^*}^{(\eta)}\\ &\quad= \widetilde{\beta}_{i^*,j_\eta^*}^{(\eta)}\cdot\left((\mathbf{Z}_{0,i^*}^{(\eta)} - c_{i^*}^{(\eta)})\circ(\mathbf{Z}_{0,i^*}^{(\eta)} - 2c_{i^*}^{(\eta)})\circ\mathbf{U}_{2,i^*} - \mathbf{Z}_{0,i^*}^{(\eta)}\circ\mathbf{U}_{3,i^*} + \mathbf{U}_{4,i^*}\right)\end{aligned}$$

By substituting the commitment openings for $\mathbf{U}_{2,i^*}$, $\mathbf{U}_{3,i^*}$, $\mathbf{U}_{4,i^*}$ with the appropriate choice of $\eta\in[3]$ and $\mathbf{Z}_{0,i^*}^{(\eta)} = \mathsf{M}_{1,i^*} + c_{i^*}^{(\eta)}\cdot\mathsf{M}_{2,i^*}$, we further obtain

$$\begin{aligned}&\left(\mathsf{M}_{1,i^*}\circ\mathsf{M}_{1,i^*}\circ\mathsf{M}_{2,i^*} - \mathsf{M}_{1,i^*}\circ\mathsf{M}_{3,i^*} + \mathsf{M}_{4,i^*}\right)\\ &\quad+ c_{i^*}^{(\eta)}\cdot\left((\mathsf{M}_{1,i^*}\circ(2\mathsf{M}_{2,i^*} - 3) - \mathsf{M}_{3,i^*})\circ\mathsf{M}_{2,i^*}\right)\\ &\quad\quad+ \left(c_{i^*}^{(\eta)}\right)^2\cdot\left(\mathsf{M}_{2,i^*}\circ(\mathsf{M}_{2,i^*} - 1)\circ(\mathsf{M}_{2,i^*} - 2)\right) = \mathbf{0}\ (\text{over } R_{q'}).\end{aligned}$$

Since this degree two polynomial evaluates to zero on three distinct $(c_{i^*}^{(\eta)})_{\eta\in[3]}\subset\mathbb{Z}_{q'}$, we must have $\mathsf{M}_{2,i^*}\circ(\mathsf{M}_{2,i^*}-1)\circ(\mathsf{M}_{2,i^*}-2) = \mathbf{0}$ over $R_{q'}$. Applying the NTT transform, this implies $\mathsf{NTT}(\mathsf{M}_{2,i^*})\in\{0,1,2\}^{dk_3\times L}$. Finally, from the verification equation regarding $\mathbf{W}_{i^*}$, we have

$$\mathsf{Rot}(\mathbf{b}_0)\left(\mathsf{NTT}(\mathbf{Z}_{0,i^*}^{(1)}) - \mathsf{NTT}(\mathbf{Z}_{0,i^*}^{(2)})\right) = (c_{i^*}^{(1)} - c_{i^*}^{(2)})\cdot\Phi(\mathbf{t}_1)\ \ (\text{over } \mathbb{Z}_{q'}).$$

Plugging in $\mathbf{Z}_{0,i^*}^{(\eta)} = \mathsf{M}_{1,i^*} + c_{i^*}^{(\eta)}\cdot\mathsf{M}_{2,i^*}$ and dividing by $(c_{i^*}^{(1)} - c_{i^*}^{(2)}) \neq 0$ (over $\mathbb{Z}_q$), we obtain

$$\mathsf{Rot}(\mathbf{b}_0)\mathsf{NTT}(\mathsf{M}_{2,i^*}) = \Phi(\mathbf{t}_1)\ \ (\text{over } \mathbb{Z}_{q'}),$$

which is equivalent to $\mathbf{b}_0\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*})) = \mathbf{t}_1$. Since we established that $\mathsf{NTT}(\mathsf{M}_{2,i^*})\in\{0,1,2\}^{dk_3\times L}$, this implies that $\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*}))\in R^{k_3\times L}$ consists of $\{0,1,2\}$-coefficient polynomials as desired.

Part (C). If $\mathsf{Extract}_{\mathsf{ss}}$ has yet to output anything, then it has computed from Part (A), $(\beta' - \widehat{\beta}')$ and $(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'})$ such that Eq. (14) holds but $(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'})/(\beta' - \widehat{\beta}')^{-1}$ does not consist of $\{0,1,2\}$-coefficient polynomials. Moreover, from Part (B), it has computed M_{2,i^*} such that $\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*}))$ consists of $\{0,1,2\}$-coefficient polynomials and $\mathbf{t}_1 = \mathbf{b}_0\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*}))$. Combining the two, we get

$$\mathbf{b}_0\underbrace{\left((\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}) - (\beta' - \widehat{\beta}')\cdot\Phi^{-1}(\mathsf{NTT}(\mathsf{M}_{2,i^*}))\right)}_{=:\mathbf{S}'} = \mathbf{0}\ (\text{over } R_{q'}),$$

Assume $\mathbf{S}' = \mathbf{0}$ over $R_{q'}$. Then, $\left(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\right) = (\beta' - \widehat{\beta}') \cdot \Phi^{-1}\left(\mathsf{NTT}\left(\mathsf{M}_{2,i^*}\right)\right)$ over $R_{q'}$. Since both sides consist only of small elements, this equation holds over $\mathbb{Z}$. Thus, it also holds over R_q. Since $(\beta' - \widehat{\beta}')$ is invertible over R_q, we can divide both sides to obtain $\left(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\right)/(\beta' - \widehat{\beta}')^{-1} = \Phi^{-1}\left(\mathsf{NTT}\left(\mathsf{M}_{2,i^*}\right)\right)$ over R_q. However, this contradicts what we have established in Part (A); $\left(\mathbf{Z}^{(1,0)'} - \widehat{\mathbf{Z}}^{(1,0)'}\right)/(\beta' - \widehat{\beta}')^{-1}$ does not consist of $\{0,1,2\}$-coefficient polynomials. Therefore, we must have $\mathbf{S}' \neq \mathbf{0}$ over $R_{q'}$. Since we established that $\mathbf{S}'$ has a small norm, $\mathsf{Extract}_{\mathsf{ss}}$ simply outputs $\mathbf{S}' \neq \mathbf{0}$ as a solution to the $\mathsf{MSIS}_{d,1,k_3,2(B_{\mathbf{Z}'}+B_c\delta^{\mathsf{gap}}),q'}$ problem for $\mathbf{b}_0$. □

$\mathsf{Multi\text{-}Extract}(1^\lambda, \mathsf{Q}_\mathsf{H}, \mathsf{Q}_\mathsf{S}, 1/\mu, \tau, \mathsf{X}, \pi^\mathsf{m})$

$\tau = (f, v) \in [-B_{\mathsf{NTRU}}, B_{\mathsf{NTRU}}]^d$ such that $p \cdot v \cdot f^{-1} = H \in \widetilde{\mathsf{crs}}^\mathsf{m}_{\mathsf{NIZK}}$,
$\mathsf{X} := (\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1), \mathsf{com} := \mathbf{T}) \in R_{q'}^{k_3} \times R_q^{k_3} \times (R_{q'}^L \times R_q^L)$,
$\pi^\mathsf{m} := ((\mathbf{u}_{0,i}, (\mathbf{U}_{k,i})_{k\in[4]}, \mathbf{V}_i)_{i\in[\tau]}, v', \mathbf{V}', \mathbf{c}_1, (\mathbf{Z}_{0,i})_{i\in[\tau]}, \mathbf{c}_2, (\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, (f'_b, \mathbf{F}'_b)_{b\in[2]})$

// Recover (a_1, a_2) by running $\mathsf{Verify}^{\mathsf{H}_\mathsf{m}}(\mathsf{crs}^\mathsf{m}_{\mathsf{NIZK}}, \mathsf{X}, \pi^\mathsf{m})$ (cf. Fig. 5)
// and proceed as follows using the computed $(\mathbf{w}'_1, \mathbf{w}'_2, (\overline{\mathbf{V}}_{i,j})_{(i,j)\in[\tau]\times[\tau']}, \overline{v}', \overline{\mathbf{V}}')$

$(\mathsf{GoodChall}_1, \mathsf{GoodChall}_2) := \{\beta'\} \times \emptyset$
$\mathsf{BadChall}_1 := \{\beta'\}$
$\mathsf{resp} := \left((\mathbf{Z}_{i,j}, \mathbf{F}_{1,i,j}, \mathbf{F}_{2,i,j})_{(i,j)\in[\tau]\times[\tau']}, \mathbf{Z}', \zeta, f'_1, f'_2, \mathbf{F}'_1, \mathbf{F}'_2\right)$
$\mathsf{trans} := (a_1, \mathbf{c}_1 = (c_i)_{i\in[\tau]}, a_2, \mathbf{c}_2 = (\boldsymbol{\beta} = (\beta_{i,j})_{(i,j)\in[\tau]\times[\tau']}, \beta'), \mathsf{resp})$
$t := 0$
While $t \leq T_{\mathsf{max}} := \frac{\lambda \cdot 2\mathsf{Q}_\mathsf{H}}{\mu}$ $\vee$ $|\mathsf{GoodChall}_1| \leq 1$
  $\beta'_t \xleftarrow{\$} C_{\mathsf{ham}} \backslash \mathsf{BadChall}_1$
  $\zeta_t := f^{-1} \cdot (f \cdot (\overline{v}' + \beta'_t \cdot v') \mod Q) \mod p$
  $\mathbf{Z}'_t := f^{-1} \cdot (f \cdot (\overline{\mathbf{V}}' + \beta'_t \cdot \mathbf{V}') \mod Q) \mod p$
  If $\|\zeta_t\|_2 < B \wedge \|\mathbf{Z}'_t\|_2 < B_{\mathbf{Z}} \wedge \mathbf{w}'_1 = \mathbf{b}_0\mathbf{Z}'_t - \beta'_t \cdot \mathbf{t}_1$
    $\wedge\ \mathbf{w}'_2 = \mathbf{b}_1\mathbf{Z}'_t\Delta + [\zeta_t|0|\cdots|0] - \beta'_t \cdot \mathbf{t}_2\Delta$ **then**
    $\mathsf{GoodChall}_1 \leftarrow \mathsf{GoodChall}_1 \cup \{\beta'_t\}$
  Else
    $\mathsf{BadChall} \leftarrow \mathsf{BadChall} \cup \{\beta'_t\}$
  $t \leftarrow t + 1$
If $|\mathsf{GoodChall}_1| = 1$ **then**
  return $= \perp$
For $i' \in [\tau]$
  For $j' \in [\tau']$
    For $\beta \in C_X$
      $\mathbf{Z}_{\beta,i',j'} := f^{-1} \cdot (f \cdot (\bar{\mathbf{V}}_{i',j'} + \beta \cdot \mathbf{V}_{i'}) \mod Q) \mod p$
      If $\|\mathbf{Z}_{\beta,i',j'}\|_2 < B_{\mathbf{Z}}$ **then**
        $\mathsf{GoodChall}_2[i'] \leftarrow \mathsf{GoodChall}_2[i'] \cup \{\beta\}$
  If $|\mathsf{GoodChall}_2[i']| \geq 2$ **then**
    For $\left(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\right)_{\eta\in[3]} \subseteq \mathsf{GoodChall}_2[i']$ s.t. $\forall \eta \in [3], \beta^{(\eta,0)}_{i',j_\eta} \neq \beta^{(\eta,1)}_{i',j_\eta}$
      $\mathsf{W} \leftarrow \mathsf{Extract}_{\mathsf{ss}}\left(\tau, \mathsf{X}, \left(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\right)_{\eta\in[3]}, \mathsf{GoodChall}_1\right)$
      If $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}^\mathsf{m}_{\mathsf{gap}}$ **then**
        return W
Else return $\perp$

Figure 8: The multi-proof straight-line extractor. We assume without loss of generality that π^m is a valid proof.

We are finally ready to finish the proof of Theorem 4.6. The full description of our multi-proof extractor Multi-Extract is provided in Fig. 8. The goal of Multi-Extract is to collect the necessary inputs to invoke $\mathsf{Extract}_{\mathsf{ss}}$ defined in Lemma 4.10.

At a high level, 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 4.10. In the following argument, we condition on $|\mathsf{GoodChall}_1| \neq 1$; that is $\mathsf{GoodChall}_1 = \{\beta', \widehat{\beta}'\}$ for some $\widehat{\beta}' \neq \beta'$.

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 $\|\mathbf{Z}_{\beta,i',j'}\|_2 < B_{\mathbf{Z}}$. Specifically, $\mathsf{GoodChall}_2[i']$ is guaranteed to collect all the $\beta \in C_X$ for which there exists a corresponding valid transcript (and some β such that $\|\mathbf{Z}_{\beta,i',j'}\|_2 < B_{\mathbf{Z}}$ but does not have an associated valid transcript).

Finally, Multi-Extract is ready to run $\mathsf{Extract}_{\mathsf{ss}}$. It runs through all three pairs of distinct challenges $\big(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\big)_{\eta\in[3]}$ from $\mathsf{GoodChall}_2[i']$ (where each pair can be the same), and executes $\mathsf{Extract}_{\mathsf{ss}}\big(\tau, \mathsf{X}, \big(\beta^{(\eta,0)}_{i',j_\eta}, \beta^{(\eta,1)}_{i',j_\eta}\big)_{\eta\in[3]}, \mathsf{GoodChall}_1\big)$. We show that with non-negligible probability, one of the set of inputs to $\mathsf{Extract}_{\mathsf{ss}}$ must be in the specified form explained in Lemma 4.10. Thus, assuming the MSIS problem is difficult, $\mathsf{Extract}_{\mathsf{ss}}$ (and thus Multi-Extract) extracts a witness W in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ as desired.

It remains to analyze the success probability and runtime of Multi-Extract. We first analyze the success probability. From Lemma 4.7, with probability at least $\frac{\mu}{2} - \mathsf{negl}(\lambda)$, we have $|\Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq \frac{\mu}{2\mathsf{Q}_{\mathsf{H}}} \cdot |C_{\mathsf{ham}}|$. By the above argument, we have $\beta'_t \in \mathsf{GoodChall}_1$ if and only if $\beta'_t \in \Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$. Therefore, if $T_{\mathsf{max}} = \frac{\lambda \cdot 2\mathsf{Q}_{\mathsf{H}}}{\mu}$, the probability that $|\mathsf{GoodChall}_1| = 2$ for all $k \in [\mathsf{Q}_{\mathsf{S}}]$ is at least $1 - \mathsf{Q}_{\mathsf{S}} \cdot 2^{-\lambda}$. We also have that for all $k \in [\mathsf{Q}_{\mathsf{S}}], |\Gamma_{1,i}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq 3$. Then, by definition of $\Gamma_{1,i}$ and our above argument, $\mathsf{Extract}_{\mathsf{ss}}$ must be invoked on the required inputs specified by Lemma 4.10 for at least one $i' \in [\tau]$. That is, for some $i' \in [\tau]$, we have $|\Gamma_{1,i'}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq 3$, and for each $\bar{c}_i \in \Gamma_{1,i'}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})$, Multi-Extract succeeds in extracting all the possible $\beta_{i',j'}$'s that can be included in a valid transcript, which by definition of $\Gamma_{1,i'}$ is more than two. Therefore, by Lemmata 4.7 and 4.10, with probability at least $\frac{\mu}{2} - \mathsf{negl}(\lambda)$, Multi-Extract extracts a witness $\mathsf{W} \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ or an MSIS solution. Assuming the MSIS problem is difficult, Multi-Extract extracts a witness $\mathsf{W} \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ with probability at least $\frac{\mu}{2} - \mathsf{negl}(\lambda)$ as desired.

We finish by analyzing the runtime of Multi-Extract. Multi-Extract takes at most time $T_{\mathsf{max}} \cdot \mathsf{poly}_{\mathsf{NTRU}}(\lambda)$ when running through the challenges in C_{ham}, where $T_{\mathsf{max}} = \frac{\lambda \cdot 2\mathsf{Q}_{\mathsf{H}}}{\mu}$ and $\mathsf{poly}_{\mathsf{NTRU}}(\lambda)$ is roughly the time it takes to perform an NTRU decryption. Moreover, it takes $\tau \cdot \tau' \cdot \mathsf{poly}_{\mathsf{NTRU}}(\lambda)$ to compute $\mathsf{GoodChall}_2$, and since $\mathsf{GoodChall}_2[i']$ has size at most $|C_X| = 2d$ for each i', Multi-Extract executes $\mathsf{Extract}_{\mathsf{ss}}$ at most $(2d)^6$-times. Since it takes $\mathsf{poly}_{\mathsf{Extract}_{\mathsf{ss}}}(\lambda)$ to run $\mathsf{Extract}_{\mathsf{ss}}$, which is in particularly independent of the runtime of the adversary $\mathcal{A}$, the total runtime of Multi-Extract is bounded by $(\frac{\lambda \cdot 2\mathsf{Q}_{\mathsf{H}}}{\mu} + \tau \cdot \tau') \cdot \mathsf{poly}(\lambda) + \mathsf{poly}_{\mathsf{Extract}_{\mathsf{ss}}}(\lambda) \cdot \tau \cdot (2d)^6$. Hence, the runtime of Multi-Extract is upper bounded by $\frac{\mathsf{Q}_{\mathsf{H}}}{\mu} \cdot p(\lambda)$ for some polynomial $p(\lambda)$ independent of $\mathcal{A}$ as desired. □

4.4 Optimization in the Classical ROM

In the context of blind signatures, we notice that we do not require the full straight-line extraction capability of our multi-proof $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$. Specifically, we can reduce the proof size by removing the Katsumata transform [Kat21] applied to the exact proof of Bootle et al. [BLS19]. These components are the first block of texts highlighted in gray in Fig. 4, where the prover commits to $(\mathbf{E}_i, (\mathbf{S}_{i,j})_{j\in[\tau']})_{i\in[\tau]}$ by the NTRU commitment/encryption scheme.

To explain why we can remove this part, we first recall Lemma 4.10 where we constructed a restricted special sound extractor $\mathsf{Extract}_{\mathsf{ss}}$. As we mentioned during in the proof of Lemma 4.10 $\mathsf{Extract}_{\mathsf{ss}}$ can only extract a valid witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ during Part (A). After Part (A), we know $\mathsf{Extract}_{\mathsf{ss}}$ can only extract an MSIS

solution. Moreover, notice in Part (A) that we never used components related to the exact proof of Bootle et al. [BLS19]. Namely, focusing only on Part (A), we can think of another extractor $\mathsf{Extract}'_{\mathsf{SS}}$ that only takes $(\mathsf{trans}^{(1,0)}, (\beta', \widehat{\beta}'))$ as input, rather than $\left(\mathsf{trans}^{(1,0)}, \left(\beta^{(\eta,0)}_{i^*,j^*_\eta}, \beta^{(\eta,1)}_{i^*,j^*_\eta}\right)_{\eta\in[3]}, (\beta', \widehat{\beta}')\right)$.

Now, looking back at our blind signature, the only moment we used the straight-line extraction property was during in the proof of one-more unforgeability in Theorem 3.5. Taking a closer look, the only reason why we required a straight-line extractor was because the simulation needed to extract the witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ to perform the simulation. If it failed to extract a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$, then there is no point for the reduction to continue simulating the rest of the game to the adversary.

Combining the two observations, it is easy to see that during the proof of one-more unforgeability, the reduction only needs the capability of running $\mathsf{Extract}'_{\mathsf{SS}}$ rather than $\mathsf{Extract}_{\mathsf{ss}}$. In case $\mathsf{Extract}'_{\mathsf{SS}}$ fails, then by the observation made during in the proof of Lemma 4.10, we know that $\mathsf{Extract}_{\mathsf{ss}}$ would have failed to output a witness for $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ as well. Therefore, once $\mathsf{Extract}'_{\mathsf{SS}}$ fails, the reduction terminates the simulation of the one-more unforgeability proof and then switches to extract an MSIS solution from the adversary via *rewinding* as it is done in the original proof of Bootle et al. [BLS19].

In summary, the reduction only needs to collect the inputs required to run Part (A) of $\mathsf{Extract}_{\mathsf{ss}}$ in a straight-line fashion. If Part (A) fails, then it can resort to rewinding-type extractions. This allows to remove all the components related to the Katsumata transform [Kat21] applied to the exact proof of Bootle et al. [BLS19], which is a huge efficiency gain. We chose to provide the NIZK with full straight-line extraction capability since it is not obvious if this idea works against quantum adversaries in the QROM. This is mainly because rewinding quantum adversaries is generally a non-trivial process and the reduction requires to also extract the MSIS solution without rewinding. We leave it as an interesting question whether this optimization applies in the quantum setting.

4.5 Putting Everything Together

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

Table 2: Concrete parameters for our scheme.

For reference, we give, in Figure 9, an instantiation of the blind signature of Section 3 using the primitives defined in this section. Note that instead of considering a straight line extractor which relies on RLWE we will consider one that relies on MLWE, e.g. we will encrypt randomness $\mathbf{R}$ by computing $\mathbf{V}' = \sum_{j=1}^{\kappa} H_j \mathbf{D}'_{1,j} + p\mathbf{D}'_2 + \mathbf{R}$ this way we can argue zero-knowledge using MLWE in dimension κ and use a much smaller modulus Q. To set parameters we consider all the constraints listed in Annexe D. For 128 bits of security we use a root-Hermite factor of $\delta_0 = 1.00454$ to be consistent with the LWE-Estimator from [APS15]. When

$\Pi_{\mathsf{BS}} : \mathcal{S}^{\mathsf{H}}(\mathsf{vk} = (\mathbf{a}_1, u), \mathsf{sk} = T_{\mathbf{a}_1})$		$\mathcal{U}^{\mathsf{H}}(\mathsf{vk}, \mathsf{M})$
$(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2) := \mathsf{H}(\text{`crs'}\|0)$, where $(\mathbf{a}_1, \mathbf{a}_2, u) \in R_q^{k_1} \times R_q^{k_2} \times R_q$		
$\mathsf{crs}_{\mathsf{com}} := (\mathbf{b}_0, \mathbf{b}_1) \in R_{q'}^{k_3} \times R_q^{k_3}$		
		$h := \mathsf{H}_{\mathsf{m}}(M)$
		$\mathbf{R} \xleftarrow{\$} [-1,1]^{k_3 \times k_2}_{\mathsf{coeff}}$
	$\xleftarrow{(\mathsf{com}, \pi^{\mathsf{m}})}$	$\mathsf{com} := \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t} \end{bmatrix} := \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \mathbf{R} + \begin{bmatrix} \mathbf{0} \\ h \cdot \mathbf{g} \end{bmatrix}$
		$\pi^{\mathsf{m}} \xleftarrow{\$} \mathsf{Prove}^{\mathsf{H}_{\mathsf{m}}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), (h, \mathbf{R} + \mathbb{1}))$
If $\mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}), \pi^{\mathsf{m}}) = \perp$ **then return** $\perp$		
$\mathbf{e} \xleftarrow{\$} \mathsf{SampleLeft}(\mathbf{a}_1, [\mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1], u, \mathbf{T}_{\mathbf{a}_1}, \sigma)$	$\xrightarrow{\mathbf{e}}$	
		$(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) := \mathbf{e} \in R_q^{k_1} \times R_q^{k_2} \times R_q^{k_3}$
		Check:
		$\forall i \in [3], \|\mathbf{e}_i\|_2 \leq B^{\mathcal{S}}_{\Sigma,i}$
		and $[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1] \cdot \mathbf{e}^\top = u$
		$\tilde{\mathbf{e}}^\top := \begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \\ \mathbf{R}\mathbf{e}_2^\top + \mathbf{e}_3^\top \end{bmatrix}$
		$\pi^{\mathsf{s}} \xleftarrow{\$} \mathsf{Prove}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}_1, \mathbf{u}, h), \tilde{\mathbf{e}})$
		return π^{s}

Figure 9: Blind signature protocol using the building blocks of Section 4. In above, $\mathbb{1}$ denotes the all one matrix.

estimating the hardness of the $\mathsf{MSIS}_{d,n,k,B,q}$ problem we use the root-Hermite factor defined as

$$\delta_0 := (Bq^{\frac{n}{k}})^{\frac{1}{dk}}$$

It should be noted that by ignoring some columns in the MSIS instance one can consider any dimension $k' \leq k$ when computing the root-Hermite factor. We thus use the value obtained by considering the maximal δ_0 obtained when varying the dimension k, which is

$$\delta_0 = 2^{\frac{\log^2 B}{4dn \log q}}.$$

We first set the modulus q as well as the dimensions (d, k_1, k_2, k_3) by considering the constraints on the hiding property of the commitment, the unforgeability of the blind signature and the quality of the three corresponding trapdoors, while taking into account that R must only split once modulo q to ensure that small messages will be invertible. The parameters of Figure 2 give $\delta_0 = 1.00262$ and $\delta_0 = 1.00443$ for the corresponding $\mathsf{MLWE}_{d,2,k_3-2,S_3,q}$ and $\mathsf{MSIS}_{d,1,k_1+k_3,B_{\mathsf{MSIS}},q}$ problems resulting in respectively 242 and 131 bits of security, where S_3 is the uniform distribution over $[-1,1]^{\mathsf{coeff}} \subset R_q$. We then fix p and Q so that decryption of the NTRU encryption (used by the straight-line extractor) always succeeds, consequently we set κ and k_4 so that the MLWE instances corresponding to the zero-knowledge property of the multi-proof extractable NIZK are hard. With the parameters of Table 2 we obtain a root-Hermite factor of $\delta_0 = 1.00286$ for the aforementioned $\mathsf{MLWE}_{d,1,\kappa,\gamma_{\mathbf{D}}/\gamma_{\mathbf{D}'}/\gamma'_{\vec{d}},Q}$ instances corresponding to 218 bits of security. We can then set q', τ, τ' to guarantee that the multi-proof extractable NIZK is sound, while making sure that R splits completely modulo q', we obtain $\mathsf{MSIS}_{d,1,k_4,16B_{\mathbf{Z}},q'}$ and $\mathsf{MSIS}_{d,1,k_3,2(B_{\mathbf{Z}'}+B_c\delta^{\mathsf{gap}}),q'}$ instances with $\delta_0 \leq 1.0021$, corresponding to 320 bits of security. We also check that the hiding property of the commitment holds for the modulus q' which is clear since $q' < q$.

We recall that the matrix $[\mathbf{a}_1 \mid \mathbf{a}_2 + h\mathbf{g} \mid \mathbf{b}_1]$ contains two one elements corresponding to the two NTRU instances and a zero element in $\mathbf{b}_1$. Using the technique of Bai-Galbraith [BG14] we can reduce the dimension of the signature by 2. We consider that Gaussians can be encoded in $\log(2\sigma)$ bits by using the encoding of

e.g. [PFH+18]. The size of the resulting signature is

102.6 KB.

We note that the first flow does not need to explicitly contain $\mathbf{t}$ since it is already part of the zero-knowledge proof. We get a first flow of size 34 MB, but considering the optimization presented in Section 4.4 we can reduce this first flow to 851 KB.

Possible optimizations. We first consider optimizations to obtain a smaller signature size. As one can observe reading this section, the hardness of the various problems given varies from 128 bits to more than 300 bits of security. Ideally we would like to reduce the appropriate parameters to obtain problems which all give similar security guarantees and smaller signatures. We could reduce the signature size by reducing k_1 and k_3 to get a tighter MLWE security than the 242 bits given above, however taking $k_1 = 2$ or $k_3 = 3$ lowers this security directly to less than 100 bits. We could circumvent this issue by 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 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 [CPS+20] and seems nontrivial to obtain. Another solution would be to lower the size of the randomness $\mathbf{R}$ to get an MLWE instance of hardness around 128 bits, but the analysis of such a very sparse randomness has not been studied well enough to have reasonable security estimates (for example the LWE-Estimator from [APS15] gives 141 bits of security for a standard deviation of 10^{-4} which in practice would clearly be insecure since the matrix $\mathbf{R}$ would be all zeroes with overwhelming probability). Even assuming we could use a very sparse randomness $\mathbf{R}$ this would only slightly improve parameters since the bound δ^{gap} would be unchanged. To get a real improvement on the multi-proof extractable NIZK, we would need to additionally prove the sparseness of $\mathbf{R}$, which we could consider by proving statements about the hamming weight of $\mathbf{R}$ but that would make the protocol much more complicated. 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 ZKP. In particular Esgin et al. [ENS20] successfully divide the size of the proof of Bootle et al. [BLS19] nearly by a factor 8. In all likelihood using the same proof would give the same improvement and bring the size of the first flow down to around 120 KB. However using this zero-knowledge proof is not completely straightforward as extraction is more complicated and the arguments used in Lemma 4.10 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.

5 Security in the QROM

In this section, we show that our blind signature Π_{BS} in Section 4 is also secure in the QROM. In particular, we show the following three items in the subsequent subsections.

1. The (semi-)generic construction in Section 3 is also secure in the "QROM".
2. The single-proof extractable NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ provided in Section 4.2 is also secure against a "QPT" adversary.
3. The multi-proof extractable NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ provided in Section 4.3 is also secure against a "QPT" adversary.

5.1 Item 1: QROM Security of the Generic Construction

The blind signature Π_{BS} in Section 3 can be shown to be secure in the QROM following a similar proof assuming the underlying NIZKs are secure against QPT algorithms. The main noticeable difference lies in the proof of the one-more-unforgeability game. In the classical setting, the challenger guessed the hashed message h_{j^*} included in the forgery (see Game_4 of Theorem 3.5) with probability $1/\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}$ but the same naive argument

no longer holds in the QROM since the probability that the guess succeeds becomes $1/|R_q| \ll 1/2^\lambda$. Note that the previous proof will not hold even under the subexponential hardness of the MLWE problem since the complexity leveraging we need to perform depends on the parameter used by the MLWE problem.

We first show that Π_{BS} is quantumly blind under malicious keys.

Theorem 5.1. *The blind signature Π_{BS} in Section 3.2 is quantumly blind under malicious keys if the commitment scheme Π_{Com} is quantumly hiding, and the two NIZKs $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ 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 zero-knowledge.*

Proof Sketch. Assuming the underlying NIZKs are quantumly zero-knowledge, the proof of blindness under malicious keys is almost identical to the classical case. The only difference is that we modify the challenger to use $2\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}$-wise independent hash functions with appropriate domains and codomains to implement the QROs $\mathsf{H}_{\mathsf{crs}}/\mathsf{H}_{\mathsf{M}}/\mathsf{H}_{\mathsf{s}}/\mathsf{H}_{\mathsf{m}}$, respectively, where $\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}/\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}/\mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}/\mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}$ are the respective numbers of random oracle queries performed by the adversary. By Lemma 2.23, this produces the same distribution to the adversary, while the challenger's runtime slightly increases since it needs to compute the Q-wise independent hash functions. The reason for this modification is so that the adversary against the hiding of Π_{Com} and the zero-knowledge of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ and $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ can efficiently simulate the challenger. Observe that unlike a classical RO, a QRO cannot be lazily simulated since the adversary may query the entire input space in a superposition. Other than this modification, the proof is exactly identical to that of Theorem 3.4. □

We next show that Π_{BS} is quantumly one-more-unforgeable.

Theorem 5.2. *The blind signature Π_{BS} is quantumly one-more unforgeable if the two NIZKs $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ 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. The high level structure of the proof remains the same as for the classical case but there are several subtle differences. Below, we provide the full proof for completeness.

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 and $\mathcal{C}_i$ as the challenger in Game_i.

Game_1 : This is the real one-more unforgeability game. By definition, we have

$$\Pr[\mathsf{E}_1] = \epsilon.$$

Game_2 : In this game, the challenger uniformly samples $2\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{s}}}/2\mathsf{Q}_{\mathsf{H}_{\mathsf{m}}}$-wise independent hash functions to implement the QROs $\mathsf{H}_{\mathsf{crs}}/\mathsf{H}_{\mathsf{M}}/\mathsf{H}_{\mathsf{s}}/\mathsf{H}_{\mathsf{m}}$, respectively. Throughout the proof, we assume without loss of generality that these Q-wise independent hash functions are sampled from a set of all possible functions with an appropriate domain and codomain. We denote these hash functions simply as $\mathsf{H}_{\mathsf{crs}}/\mathsf{H}_{\mathsf{M}}/\mathsf{H}_{\mathsf{s}}/\mathsf{H}_{\mathsf{m}}$. By Lemma 2.23, this produces the same distribution to the adversary. Thus we have,

$$\Pr[\mathsf{E}_2] = \Pr[\mathsf{E}_1].$$

Moreover, we have $\mathsf{Time}(\mathcal{C}_2) = \mathsf{Time}(\mathcal{C}_1) + \sum_{\mathsf{str}\in\{\mathsf{crs},\mathsf{M},\mathsf{m},\mathsf{s}\}} \mathsf{Q}_{\mathsf{str}} \cdot T^{2\mathsf{Q}_{\mathsf{str}}\text{-wise}}$, where recall $T^{\mathsf{Q}\text{-wise}}$ denotes the time to evaluate a Q-wise independent hash function, which is $O(\mathsf{Q})$ for a typical choice.

Game_3 : In this game, the challenger modifies the description of the function $\mathsf{H}_{\mathsf{crs}}$. It first samples a random $2\mathsf{Q}_{\mathsf{H}_{\mathsf{crs}}}$-wise independent hash function $\mathsf{H}'_{\mathsf{crs}}$ as in the previous game and further runs the CRS simulator $\mathcal{S}_{\mathsf{crs}}$ provided by $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ and generates $(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^\lambda)$. It then sets the function $\mathsf{H}_{\mathsf{crs}}$ as

$$\mathsf{H}_{\mathsf{crs}}(x) = \begin{cases} (\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2) & \text{if } x = 0, \\ \mathsf{H}'_{\mathsf{crs}}(x) & \text{otherwise.} \end{cases} \tag{16}$$

Otherwise, the challenger proceeds identically to Game_1.

It can be checked that Game_2 and Game_3 are indistinguishable by the CRS indistinguishability in Definition 2.10. Specifically, there exists a QPT adversary $\mathcal{B}_{\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}}$ against the CRS indistinguishability such that

$$\Pr[\mathsf{E}_3] \geq \Pr[\mathsf{E}_2] - \mathsf{Adv}^{\mathsf{crs}}_{\Pi^{\mathsf{m}}_{\mathsf{NIZK}}}(\mathcal{B}_{\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}}),$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}})$ is $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_2)$. Note that $\mathcal{B}_{\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}}$ can efficiently simulate $\mathcal{C}_2$ due to the modification made in Game_2. Assuming CRS indistinguishability, we have $\Pr[\mathsf{E}_3] \geq \Pr[\mathsf{E}_2] - \mathsf{negl}(\lambda)$.

Game_4 : In this game, the challenger uses the multi-proof extractor $\mathsf{Multi\text{-}Extract}$ provided by $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ to extract a witness in $\mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$ from all the proofs included in Q_{S} first messages $(\rho_{j,1})_{j\in[\mathsf{Q}_{\mathsf{S}}]}$ submitted by $\mathcal{A}$. Specifically, when $\mathcal{A}$ submits $\rho_{j,1} = (\mathsf{com}_j, \pi^{\mathsf{m}}_j)$ to the challenger, 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 $\mathsf{X}_j = (\mathsf{crs}_{\mathsf{com}}, \mathsf{com}_j)$. We denote by $\mathsf{Abort}_{\mathsf{extract}}$ the event that there exists $j \in [\mathsf{Q}_{\mathsf{S}}]$ such that $\mathsf{W}_j \notin \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. If $\mathsf{Abort}_{\mathsf{extract}}$ occurs, the challenger aborts the game and rewrites the forgery of $\mathcal{A}$ to be $\perp$. Otherwise, it proceeds identically to Game_3. Conditioning on $\mathsf{Abort}_{\mathsf{extract}}$ not occurring, the challenger extracts $\mathsf{W}_j = (h'_j, c'_j, c_j, (\mathbf{r}_{j,i})_{i\in[k_2]}) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$. We note that the challenger does not use the extracted witness in this game.

An identical argument to the classical case (see Lemma 3.6) shows that

$$\Pr[\mathsf{E}_4] \geq \frac{1}{2} \cdot \Pr[\mathsf{E}_3] - \mathsf{negl}(\lambda).$$

In this game, the runtime of the challenger $\mathcal{C}_4$ becomes longer than that of $\mathcal{C}_3$ since it runs the multi-proof extractor $\mathsf{Multi\text{-}Extract}$. Due to Definition 2.10, we have $\mathsf{Time}(\mathcal{C}_4) = \mathsf{Time}(\mathcal{C}_3) + \mathsf{Q}^{e_1}_{\mathsf{H}_{\mathsf{m}}} \cdot \mathsf{Q}^{e_2+1}_{\mathsf{S}} \cdot \frac{1}{\mu^c} \cdot p(\lambda)$ for some constants (c, e_1, e_2) and polynomial $p(\lambda)$, where $\mu = \Pr[\mathsf{E}_3] \geq \epsilon - \mathsf{negl}(\lambda)$. Assuming ϵ is non-negligible, $\mathsf{Time}(\mathcal{C}_4)$ is bounded by a polynomial.

Game_5**:** In this game, the challenger checks if all the messages $\{\mathsf{M}_i\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$ in the adversary's forgery satisfy $\mathsf{H}_{\mathsf{M}}(\mathsf{M}_i) \neq \mathsf{H}_{\mathsf{M}}(\mathsf{M}_j)$ for $i \neq j \in [\mathsf{Q}_{\mathsf{S}} + 1]$, where we denote the event that a collision is found by $\mathsf{Abort}'_{\mathsf{guess}}$. If $\mathsf{Abort}'_{\mathsf{guess}}$ occurs, the challenger aborts the game and rewrites the forgery of $\mathcal{A}$ to be $\perp$. Otherwise, it proceeds identically to Game_4. By [Zha15], any (possibly unbounded) quantum algorithm making $\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}$ queries can find a collision with probability at most $C' \cdot (\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}} + 1)^3 / |S_{\mathsf{hash}}|$ for some universal constant C'. Therefore, we have

$$\Pr[\mathsf{E}_5] \geq \Pr[\mathsf{E}_4] - \frac{C' \cdot (\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}} + 1)^3}{|S_{\mathsf{hash}}|}.$$

Game_6 : In this game, 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 $r = \frac{2 \cdot C_0 \cdot \mathsf{Q}^3_{\mathsf{H}_{\mathsf{M}}}}{\mu'}$, where $\mu' = \Pr[\mathsf{E}_5]$ and C_0 is defiend as in Definition 2.24. 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 $\mathsf{H}_{\mathsf{M}}(x) = h_{P(x)}$. Since H_{M} is drawn from the small-range distribution with r samples from the set $D = S_{\mathsf{hash}}$, Lemma 2.25 asserts that

$$\Pr[\mathsf{E}_6] \geq \Pr[\mathsf{E}_5] - \frac{C_0 \cdot \mathsf{Q}^3_{\mathsf{H}_{\mathsf{M}}}}{r} = \frac{1}{2} \cdot \Pr[\mathsf{E}_5].$$

Here, since sampling and computing P takes time $|\mathcal{M}|$, which is in general exponential in λ, the challenger instead uses a $T^{2\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}\text{-wise}}$ independent hash function H_P to simulate P. By Lemma 2.23, $\mathsf{H}_{\mathsf{M}}(x) = h_{\mathsf{H}_P(x)}$ produces the same distribution to the adversary, and thus, the above bound on $\Pr[\mathsf{E}_5]$ remains the same. Moreover, we have $\mathsf{Time}(C_6) = \mathsf{Time}(C_5) + r \cdot \mathsf{poly}(\lambda)$, where $r = \frac{2 \cdot C_0 \cdot \mathsf{Q}^3_{\mathsf{H}_{\mathsf{M}}}}{\mu'}$, $\mu' = \Pr[\mathsf{E}_5]$, and $\mathsf{poly}(\lambda)$ is the time it takes to uniformly sample from S_{hash}. Note that the time to compute H_P does not show up explicitly since $\mathcal{C}_5$ also computes a similar hash function. Assuming μ' is non-negligible and $\mathsf{Time}(\mathcal{C}_5)$ is polynomial, $\mathsf{Time}(\mathcal{C}_6)$ is also polynomial.

Game_7 : In this game, 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 $\mathsf{W}_j = (h'_j, c'_j, c_j, (\mathbf{r}_{j,i})_{i\in[k_2]}) \in \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}}$

from the first message $\rho_{j,1}$ submitted to by $\mathcal{A}$ (conditioned on $\mathsf{Abort}_{\mathsf{extract}}$ not occurring), the challenger checks if $h'_j/c'_j \neq h_{j^*}$, where note that by definition c'_j is invertible. Moreover, at the end of the game, when $\mathcal{A}$ outputs the forgery $\{(\mathsf{M}_i, \Sigma_i)\}_{i\in[\mathsf{Q_S}+1]}$, the challenger checks if $h_{j^*} \in \{\mathsf{H}(\mathsf{M}_i)\}_{i\in[\mathsf{Q_S}+1]}$ and if $\{\mathsf{H_M}(\mathsf{M}_i)\}_{i\in[\mathsf{Q_S}+1]}$ are pairwise distinct. We denote by $\mathsf{Abort}_{\mathsf{guess}}$ the event that either of these checks do not hold, where note that event $\mathsf{Abort}_{\mathsf{guess}}$ includes event $\mathsf{Abort}'_{\mathsf{guess}}$. If $\mathsf{Abort}_{\mathsf{guess}}$ occurs, the challenger aborts and rewrites the forgery of $\mathcal{A}$ to be $\perp$. Otherwise, it proceeds identically to Game_6.

We later show in Lemma 5.3 that

$$\Pr[\mathsf{E}_7] \geq \frac{1}{2r} \cdot \Pr[\mathsf{E}_6],$$

where recall $r = \frac{2\cdot C_0 \cdot \mathsf{Q}_{\mathsf{H_M}}^3}{\mu'}$ and $\mu' = \Pr[\mathsf{E}_5]$. Finally, it can be checked that $\mathsf{Time}(\mathcal{C}_7) = \mathsf{Time}(\mathcal{C}_6)$.

Game_8 : In this game, the challenger modifies $\mathbf{a}_2$ in the output $\mathsf{H}_{\mathsf{crs}}(0) = (\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{crs}_{\mathsf{com}}, \mathbf{a}_2)$. Specifically, after it samples $j^* \xleftarrow{\$} [r]$ at the beginning of the game, it sets $\mathbf{a}_2 = \widetilde{\mathbf{a}}_2 - h_{j^*} \cdot \mathbf{g}$ where $\widetilde{\mathbf{a}}_2 \xleftarrow{\$} R_q^k$, and sets $\mathsf{H}_{\mathsf{crs}}$ as in Eq. (16). Since the distribution of $\mathbf{a}_2$ in both games are uniform over R_q^k, we have

$$\Pr[\mathsf{E}_8] = \Pr[\mathsf{E}_7].$$

Game_9 : In this game, the challenger gets rid of the trapdoor $\mathbf{T}_{\mathbf{a}_1}$ included in the secret key sk and modifies the way it samples the short vector $\mathbf{e}$ when $\mathcal{A}$ submits the first message ρ_1. We omit the details as it is defined identically to Game_6 of the classical case in Theorem 3.5. Following the same argument as in the classical case (see Lemma 3.8), there exists PPT adversaries $\mathcal{B}_{\mathsf{MLWE}}$, $\mathcal{B}'_{\mathsf{DSMR}}$ and $\mathcal{B}_{\mathsf{DSMR}}$ against the $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$, $\mathsf{DSMR}_{d,k_1,\chi_{\mathsf{DSMR}},q,1}$, and $\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}$ problems, respectively, such that

$$\begin{aligned}\Pr[\mathsf{E}_9] \geq \Pr[\mathsf{E}_8] - &\mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}}(\mathcal{B}_{\mathsf{MLWE}}) - \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}'_{\mathsf{DSMR}}) \\ &- 2 \cdot \mathsf{Adv}^{\mathsf{DSMR}_{d,k_2k_3-1,\chi_{\mathsf{DSMR}},q,1}}(\mathcal{B}_{\mathsf{DSMR}}) - \mathsf{negl}(\lambda)\end{aligned}$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{MLWE}})$, $\mathsf{Time}(\mathcal{B}'_{\mathsf{DSMR}})$, and $\mathsf{Time}(\mathcal{B}_{\mathsf{DSMR}})$ are roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_9)$. Assuming the hardness of the MLWE and DSMR problems, we have $\Pr[\mathsf{E}_9] \geq \Pr[\mathsf{E}_8] - \mathsf{negl}(\lambda)$.

At this point, the challenger in Game_9 no longer relies on a trapdoor for $\mathbf{a}_1$. Therefore, we are now ready to embed an MSIS instance in the public vectors and to simulate the view of $\mathcal{A}$ in Game_9 in order to solve the MSIS problem. Following an identical proof to the classical case (see Lemma 3.9), there exists a QPT adversary $\mathcal{B}_{\mathsf{MSIS}}$ against the MSIS problem such that

$$\mathsf{Adv}^{\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}}(\mathcal{B}_{\mathsf{MSIS}}) \geq \frac{1}{2p(\lambda)\cdot \mathsf{Q}_{\mathsf{H_s}}^e} \cdot \Pr[\mathsf{E}_9]^{c_1} - \mathsf{negl}(\lambda),$$

where $p(\lambda)$ is a polynomial, and e and c_1 are constants defined in Definition 2.9. Moreover, we have $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \leq c_2 \cdot (\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_9))$, where c_2 is also a constant defined in Definition 2.9.

Let us check that $\mathcal{B}_{\mathsf{MSIS}}$ has non-negligible advantage and runs in polynomial time to arrive at a contradiction. Collecting all the bounds, we have

$$\Pr[\mathsf{E}_9] \geq \frac{1}{8r} \cdot \Pr[\mathsf{E}_1] - \mathsf{negl}(\lambda) = \frac{\mu'}{16 \cdot C_0 \cdot \mathsf{Q}_{\mathsf{H_M}}^3} \cdot \epsilon - \mathsf{negl}(\lambda) \geq \frac{\epsilon^2}{432 \cdot \mathsf{Q}_{\mathsf{H_M}}^3} - \mathsf{negl}(\lambda),$$

where we used the fact $|S_{\mathsf{hash}}| \geq 2^\lambda$, $r = \frac{2\cdot C_0 \cdot \mathsf{Q}_{\mathsf{H_M}}^3}{\mu'}$, $\mu' = \Pr[\mathsf{E}_5]$, $\Pr[\mathsf{E}_5] \geq \frac{1}{2} \cdot \epsilon - \mathsf{negl}(\lambda)$, and $C_0 < 27$ from Lemma 2.25. This in particular implies

$$\mathsf{Adv}^{\mathsf{MSIS}_{d,1,k_1+k_2k_3,B_{\mathsf{MSIS}},q}}(\mathcal{B}_{\mathsf{MSIS}}) \geq \frac{1}{2p(\lambda)\cdot \mathsf{Q}_{\mathsf{H_s}}^e} \cdot \left(\frac{\epsilon^2}{432 \cdot \mathsf{Q}_{\mathsf{H_M}}^3}\right)^{c_1} - \mathsf{negl}(\lambda)$$

which is non-negligible by assumption. Moreover, we have $\mathsf{Time}(\mathcal{C}_9) \approx \cdots \approx \mathsf{Time}(\mathcal{C}_6)$, $\mathsf{Time}(\mathcal{C}_6) = \mathsf{Time}(\mathcal{C}_5)+r\cdot\mathsf{poly}(\lambda)$, $\mathsf{Time}(\mathcal{C}_5) = \mathsf{Time}(\mathcal{C}_4)+\mathsf{Q}_{\mathsf{H_m}}^{e_1}\cdot\mathsf{Q}_{\mathsf{S}}^{e_2+1}\cdot\frac{1}{\mu^c}\cdot p(\lambda)$, $\mathsf{Time}(\mathcal{C}_4) = \mathsf{Time}(\mathcal{C}_1)+\sum_{\mathsf{str}\in\{\mathsf{crs},\mathsf{M},\mathsf{m},\mathsf{s}\}}\mathsf{Q}_{\mathsf{str}}\cdot T^{2\mathsf{Q}_{\mathsf{str}}\text{-}\mathsf{wise}}$, where $\mu = \Pr[\mathsf{E}_3] \geq \epsilon - \mathsf{negl}(\lambda)$ and "$\approx$" hides an insignificant blow up in the runtime. Since $\mathsf{Time}(\mathcal{A})$ can be assumed to be larger than $\mathsf{Time}(\mathcal{C}_1)$, we have

$$\mathsf{Time}(\mathcal{C}_9) \approx \mathsf{Time}(\mathcal{A}) + \frac{4\cdot C_0\cdot \mathsf{Q}_{\mathsf{H_M}}^3}{\epsilon}\cdot\mathsf{poly}(\lambda) + \frac{\mathsf{Q}_{\mathsf{H_m}}^{e_1}\cdot\mathsf{Q}_{\mathsf{S}}^{e_2+1}}{\epsilon^c}\cdot p(\lambda) + \sum_{\mathsf{str}\in\{\mathsf{crs},\mathsf{M},\mathsf{m},\mathsf{s}\}}\mathsf{Q}_{\mathsf{str}}\cdot T^{2\mathsf{Q}_{\mathsf{str}}\text{-}\mathsf{wise}},$$

where recall $\mathsf{poly}(\lambda)$ is the time it takes to uniformly sample from S_{hash}. Assuming ϵ is non-negligible, $\mathsf{Time}(\mathcal{C}_9)$ is polynomial. Combining this with $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) \leq c_2\cdot(\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_9))$ for constant c_2, we have $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}})$ is polynomial as desired. Since this implies a QPT adversary for the MSIS problem with non-negligible advantage, we arrive at a contradiction.

To complete the proof of the main theorem, it remains to prove the following Lemma 5.3.

Lemma 5.3. *We have* $\Pr[\mathsf{E}_7] \geq \frac{1}{2r}\cdot\Pr[\mathsf{E}_6]$.

Proof. Let us analyze $\Pr[\mathsf{Abort}_{\mathsf{guess}}]$. Notice the worst case is achieved when $\mathcal{A}$ outputs a forgery $\{(\mathsf{M}_i,\Sigma_i)\}_{i\in[\mathsf{Q_S}+1]}$, where $\{h'_j/c'_j\}_{j\in[\mathsf{Q_S}]} \subset \{\mathsf{H_M}(\mathsf{M}_i)\}_{i\in[\mathsf{Q_S}+1]}$. Conditioned on event $\mathsf{Abort}'_{\mathsf{guess}}$, we are guaranteed that $\{\mathsf{H_M}(\mathsf{M}_i)\}_{i\in[\mathsf{Q_S}+1]}$ is of size $\mathsf{Q_S}+1$. Therefore, since j^* is distributed uniformly random over $[r]$ from the view of $\mathcal{A}$, we have

$$\Pr[\mathsf{Abort}_{\mathsf{guess}}] \geq \left(1-\frac{1}{r}\right)^{\mathsf{Q_S}}\cdot\frac{1}{r} \geq \frac{1}{r} - \frac{\mathsf{Q_S}}{r^2} \geq \frac{1}{2r},$$

where we use the fact $r \geq 2\mathsf{Q_S}$. This is without loss of generality since we can always include in $\mathsf{Q_{H_M}}$ the number of hash queries performed by the challenger to run the verification algorithm, which is $\mathsf{Q_S}+1$. Since the only differences between Game_6 and Game_7 are the abort conditions, the statement follows. □

□

5.2 Item 2: QROM Security of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$

We consider the same single-proof extractable NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ for relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ provided in Section 4.2. The following is the main theorem of this section.

Theorem 5.4. *The* NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ *in Figs. 2 and 3 is quantumly single-proof extractable with* $(c_1,c_2,e) = (3,2,6)$ *and* $p(\lambda) = \mathsf{poly}(\lambda)$. *Moreover, it is quantumly zero-knowledge.*

Since zero-knowledge against quantum adversaries follows from previous work (see the discussion in Theorem 4.2), we only focus on proving single-proof extractability against quantum adversaries.

Unlike in the classical case, we cannot simply rewinding the cheating prover to extract the witness since it may disrupt the quantum prover's internal state. That is, we cannot rely on the standard argument to lower bound the probability that the prover succeeds again after being rewound with similar advantage. We thus rely on recent results on QROM secure NIZKs based on the Fiat-Shamir transform [LZ19, DFMS19, DFM20].

For completeness, we provide all the necessary tools to argue single-proof extractability of $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ in Appendix B. In short, if the underlying sigma protocol implicit in $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ (see Fig. 2) has an associated *instance generator* IGen (see Definition B.1) and a (τ,ν)-compatible separable function $\mathsf{CSF.Gen}$ (see Definition B.5), then $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ with an associated IGen is single-proof extractable with parameters $(c_1,c_2,e) = (3,2,6)$ and $p(\lambda) = (\tau-\nu)^2/4$ by Theorem B.4. This establishes Theorem 5.4.

Informally, IGen generates a statement-witness pair for which the adversary must provide a proof. In the context of blind signature, IGen is supposed to output a statement X which is distributed as D_{X}, as used in the proof of Lemma 3.9. Concretely, for $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ to be useful in our context, we define IGen as follows, where recall the parameters are defined in Table 1.

- $\mathsf{IGen}(1^\lambda)$: On input the security parameter 1^λ, it samples $\mathbf{a}_1 \xleftarrow{\$} \{1\} \times R_q^{k_1-1}, \mathbf{b} \xleftarrow{\$} R_q^{k_3}$, $\mathbf{R} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1 \times k_2}$, $u \xleftarrow{\$} R_q$, and $h \xleftarrow{\$} S_{\mathsf{hash}}$. It further samples (possibly inefficiently) a random $\tilde{\mathbf{e}} := (\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \tilde{\mathbf{e}}_3) \in R^{k_1+k_2+k_3}$ such that for all $i \in [3]$, $\|\tilde{\mathbf{e}}_i\|_2 \le B^{\mathcal{U}}_{\Sigma,i}$, and $[\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}]\, \tilde{\mathbf{e}}^\top = u$. It finally output a statement-witness pair $(\mathsf{X} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}, u, h), \mathsf{W} = \tilde{\mathbf{e}})$ such that $(\mathsf{X}, \mathsf{W}) \in (\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$.

It remains to show that the sigma protocol for relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ with an associated instance generator IGen has a (τ, μ)-compatible separable function. The following proof follows closely [LZ19, Section 5] which showed that Lyubashevsky's sigma protocol [Lyu09, Lyu12] over *non-structured* lattices has a compatible separable function. We extend their results to the case of *structured* lattices.

Lemma 5.5. *The sigma protocol for relations $(\mathcal{R}^{\mathsf{s}}, \mathcal{R}^{\mathsf{s}}_{\mathsf{gap}})$ (implicit in Fig. 2) with an associated instance generator IGen has a $(\tau(\lambda), \nu(\lambda))$-compatible separable function, where $\tau(\lambda) = 0.19$ and $\nu(\lambda) = 1/q^2$ (resp. $\nu(\lambda) = 0$) when q is odd (resp. even) assuming the hardness of the $\mathsf{MLWE}_{d,1,k_1+k_2+k_3,\chi_{\mathsf{MLWE}},q}$, where $\chi_{\mathsf{MLWE}} := D_{\mathbb{Z},\sigma}$ and $\sigma \cdot \sum_{i\in[3]} \sqrt{k_i} B_{\Sigma,i} < q/5$.*

Proof. The sigma protocol implicit in Fig. 2 uses $\alpha := w \in R_q$, $\beta := c \in S_{\mathsf{chal}}$, and $\gamma := \mathbf{z} \in R^{k_1+k_2+k_3}$ as the first, second, and third flow, respectively. We define the compatible separable function $\mathsf{CSF.Gen}$ as follows:

- $\mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha = w, \beta = c, \mathsf{mode} = \mathsf{preserving})$: When mode is $\mathsf{preserving}$, it samples $(s, \mathbf{x}) \xleftarrow{\$} \chi_{\mathsf{MLWE}} \times \chi_{\mathsf{MLWE}}^{k_1+k_2+k_3}$ and $\mathbf{d} \xleftarrow{\$} R_q$, and outputs the function $f : R^{k_1+k_2+k_3} \to \{0,1\}$ defined as

$$f(\mathbf{z}) := \left\lfloor \mathsf{coeff}_1\left((s \cdot [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}] + \mathbf{x})\, \mathbf{z}^\top + d \right) \right\rfloor_{\lfloor q/2 \rceil},$$

  where $\mathsf{coeff}_1(a)$ outputs the first coefficient a_1 of $a \in R_q$, when viewing a as a polynomial of degree d with coefficients in $\{0, 1, \cdots, q-1\}$, and $\lfloor a_1 \rfloor_{\lfloor q/2 \rceil}$ outputs $\lfloor a_1 / \lfloor q/2 \rceil \rfloor \in \{0,1\}$ for $a_1 \in \mathbb{Z}_q$ and $q \ge 2$. Here, $\lfloor x \rceil$ outputs the nearest *largest* integer, e.g., $\lfloor 1.5 \rceil = \lfloor 2.4 \rceil = 2$, and $\lfloor x \rfloor$ outputs the nearest integer x' such that $0 \le \lfloor x \rfloor - x' < 1$, e.g., $\lfloor 1 \rfloor = \lfloor 1.9 \rfloor = 1$.

- $\mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha = w, \beta = c, \mathsf{mode} = \mathsf{separating})$: When mode is $\mathsf{separating}$, it samples $\mathbf{v} \xleftarrow{\$} R_q^{k+k'+k_{\mathsf{com}}}$ and $d \xleftarrow{\$} R_q$, and outputs the function $f : R^{k_1+k_2+k_3} \to \{0,1\}$ defined as

$$f(\mathbf{z}) := \left\lfloor \mathsf{coeff}_1\left(\mathbf{v}\mathbf{z}^\top + d \right) \right\rfloor_{\lfloor q/2 \rceil}.$$

Now, for any $(\alpha, \beta) = (w, c) \in R_q \times S_{\mathsf{chal}}$, the set of all valid third flow $V_{\mathsf{X},\alpha,\beta}$ is defined as

$$V_{\mathsf{X},\alpha,\beta} := \left\{ \gamma = \mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) \;\middle|\; \begin{array}{c} [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}]\, \mathbf{z}^\top = w + c \cdot u \\ \wedge \;\; \forall i \in [3], \|\mathbf{z}_i\|_2 \le B_{\Sigma,i} \end{array} \right\} \subset R^{k_1+k_2+k_3}.$$

Below we prove the properties required from compatible separable functions defined in Definitions B.5 and B.6.

Preserving Mode. Fix any $\mathsf{X} \in \mathcal{L}_{\mathcal{R}^{\mathsf{s}}}$ and $(\alpha, \beta) = (w, c) \in R_q \times S_{\mathsf{chal}}$ such that $|V_{\mathsf{X},\alpha,\beta}| \ge 1$. For a random choice of $f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, w, c, \mathsf{preserving})$ and for any $\gamma = \mathbf{z} \in V_{\mathsf{X},\alpha,\beta}$, we have

$$\begin{aligned} f(\mathbf{z}_1, \mathbf{z}_2) &= \left\lfloor \mathsf{coeff}_1\left((s \cdot [\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \mathbf{b}] + \mathbf{x})\, \mathbf{z}^\top + d \right) \right\rfloor_{\lfloor q/2 \rceil} \\ &= \left\lfloor \mathsf{coeff}_1\left(\underbrace{s \cdot (w + c \cdot u) + d}_{=:\mathsf{pub}} + \underbrace{\mathbf{x}\mathbf{z}^\top}_{=:\mathsf{err}} \right) \right\rfloor_{\lfloor q/2 \rceil}. \end{aligned}$$

By definition of a valid third flow $\mathbf{z}$, pub is identical for $\mathbf{z} \in V_{\mathsf{X},\alpha,\beta}$. Let $\Delta = \sigma \cdot \sum_{i\in[3]} \sqrt{k_i} B_{\Sigma,i}$, which is smaller than $q/5$ by assumption. Since $\mathsf{coeff}_1(\mathsf{pub})$ is distributed uniformly random over $\mathbb{Z}_q$ for a randomly chosen f, $\mathsf{coeff}_1(\mathsf{pub})$ falls into $[\Delta, \lfloor q/2 \rceil - \Delta]$ or $[\lfloor q/2 \rceil + \Delta, q - \Delta]$ with probability $1 - 4\Delta/q \ge 1/5$. By

Lemma 2.13, we have $\left|\mathbf{x}\mathbf{z}^\top\right| \leq \sigma \cdot \sum_{i\in[3]} \sqrt{k_i} B_{\Sigma,i}$ for all $\mathbf{z} \in V_{\mathsf{X},\alpha,\beta}$ with probability at least $1 - \mathsf{negl}(\lambda)$, where $\mathbf{x} \xleftarrow{\$} \chi_{\mathsf{MLWE}}^{k_1+k_2+k_3}$. Hence, we have $|\{f(\mathbf{z}) \mid \mathbf{z} \in V_{\mathsf{X},\alpha,\beta}\}| = 1$ with probability at least $\frac{1}{5} - \mathsf{negl}(\lambda) > 0.19$.

Separating Mode. For any distinct $\mathbf{z}, \mathbf{z}' \in V_{\mathsf{X},\alpha,\beta}$, the differences between $\mathsf{coeff}_1\left(\mathbf{v}\mathbf{z}^\top\right)$ and $\mathsf{coeff}_1\left(\mathbf{v}\mathbf{z}'^\top\right)$ are uniform over $\mathbb{Z}_q$ for $\mathbf{v} \xleftarrow{\$} R_q^{k_1+k_2+k_3}$. Then, further considering the randomness over $d \xleftarrow{\$} R_q$, $(\mathsf{coeff}_1\left(\mathbf{v}\mathbf{z}^\top + d\right),$ $\mathsf{coeff}_1\left(\mathbf{v}\mathbf{z}'^\top + d\right))$ is distributed uniform over $\mathbb{Z}_q \times \mathbb{Z}_q$. Therefore, for any distinct $\mathbf{z}, \mathbf{z}' \in V_{\mathsf{X},\alpha,\beta}$, we have

$$\Pr\left[f(\mathbf{z}) = f(\mathbf{z}')\right] = 1 - \frac{2 \cdot \lfloor q/2 \rceil \cdot (q - \lfloor q/2 \rceil)}{q^2} = \begin{cases} \frac{1}{2} & \text{if } q \text{ is even} \\ \frac{1+(1/q^2)}{2} & \text{if } q \text{ is odd} \end{cases},$$

where the probability is take over $f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta, \mathsf{separating})$. Therefore, when q is even, $\nu(\lambda) = \xi(\lambda) = 0$ and when q is odd, then $\nu(\lambda) = \xi(\lambda) = \frac{1}{q^2}$ as desired.

Mode Indistinguishability. This is a direct consequence of the $\mathsf{MLWE}_{d,1,k_1+k_2+k_3,\chi_{\mathsf{MLWE}},q}$ assumption. □

5.3 Item 3: QROM Security of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$

We consider the same multi-proof extractable NIZK $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ for relations $(\mathcal{R}^{\mathsf{m}}, \mathcal{R}^{\mathsf{m}}_{\mathsf{gap}})$ provided in Section 4.3. The following is the main theorem of this section.

Theorem 5.6. *The NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ in Figs. 4 and 5 is quantumly multi-proof extractable with $(c_1, c_2, e) = (1, 2, 1)$ and $p(\lambda) = \mathsf{poly}(\lambda)$. Moreover, it is quantumly zero-knowledge.*

The proof is a consequence of the following Theorems 5.7 and 5.8.

Zero-Knowledge.

Theorem 5.7. *The NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ in Figs. 4 and 5 is quantumly zero-knowledge if the $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{d}'},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}'},Q}$, and $\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}$ problems are hard.*

Proof. Assume there exists a QPT adversary $\mathcal{A}$ with advantage ϵ, where the zero-knowledge simulator $\mathsf{Sim} := (\mathsf{Sim}_0, \mathsf{Sim}_1)$ is as defined in Fig. 7. Let us define two distributions $\mathcal{H}_0$ and $\mathcal{H}_1$ over $\mathcal{X}$ and $\mathcal{Y}$, respectively, such that $\mathcal{X} := \mathcal{R}^{\mathsf{m}} \times \{0,1\}$, where $\mathcal{R}^{\mathsf{m}}$ is the space of statement-witness pair, and $\mathcal{Y}$ is the space of all possible transcripts $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$ of the implicit 5-round interactive protocol of $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ (see Lemma 4.5 to recall their definition). Concretely, we define $\mathcal{H}_0$ and $\mathcal{H}_1$ as follows:

- When we take $\mathsf{H}_0 \xleftarrow{\$} \mathcal{H}_0$, for each $(\mathsf{X}, \mathsf{W}, b) \in \mathcal{X}$, $\mathsf{H}_0(\mathsf{X}, \mathsf{W}, b)$ is identically and independently distributed according to the distribution $D^{\not\perp}_{\mathsf{trans}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathsf{W})$ defined in the statement of Lemma 4.5.
- When we take $\mathsf{H}_1 \xleftarrow{\$} \mathcal{H}_1$, for each $(\mathsf{X}, \mathsf{W}, b) \in \mathcal{X}$, $\mathsf{H}_1(\mathsf{X}, \mathsf{W}, b)$ is identically and independently distributed according to the distribution $D_{\mathsf{sim}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X})$ defined as sampling $(\mathbf{c}_1, \mathbf{c}_2) \xleftarrow{\$} \mathbb{Z}_{q'}^{\tau} \times (C_X^{\tau \cdot \tau'} \times C_{\mathsf{ham}})$ and $(a_1, a_2, \mathsf{resp}) \xleftarrow{\$} \mathsf{Sim}_{\mathsf{int}}(\mathsf{crs}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}, \mathbf{c}_1, \mathbf{c}_2)$, and outputting $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$.

We now consider a QPT adversary $\mathcal{B}$ with oracle access to either $\mathcal{H}_0$ or $\mathcal{H}_1$ that simulates the view to $\mathcal{A}$. Notice that $\mathcal{B}$ can plug in $\mathcal{H}_0$ and $\mathcal{H}_1$ in place of the $\mathsf{GetTrans}$ algorithm in Fig. 7, where observe that the description of Sim_0 can be rewritten using $\mathsf{GetTrans}$ rather than $\mathsf{Sim}_{\mathsf{int}}$. If $\mathcal{B}$ is provided $\mathcal{H}_0$, then it perfectly simulates $(\mathsf{H}_{\mathsf{m}}, \mathsf{Prove})$ to $\mathcal{A}$.[16] On the other hand, If $\mathcal{B}$ is provided $\mathcal{H}_1$, then it perfectly simulates $(\mathsf{Sim}_0, \mathcal{S})$ to $\mathcal{A}$. $\mathcal{B}$ outputs whatever $\mathcal{A}$ output.

Then, $\mathcal{B}$ makes at most $\mathsf{Q}_{\mathsf{H\text{-}P}}$ queries and satisfies

$$\left|\Pr[\mathcal{B}^{|\mathsf{H}_0\rangle}(1^\lambda) \to 1] - \Pr[\mathcal{B}^{|\mathsf{H}_1\rangle}(1^\lambda) \to 1]\right| \geq \epsilon,$$

[16] To be precise, we require $\mathcal{H}_0$ to output $\perp$ with negligible probability to be consistent with the real-world prover that may abort with negligible probability. However, we omit this for simplicity as it makes negligible difference.

where $\mathsf{Q}_{\mathsf{H\text{-}P}}$ is the total number of queries $\mathcal{A}$ makes to H_{m} and Prove. Then, by Lemma 2.26, we can construct a QPT algorithm $\mathcal{B}'$ that distinguishes $D^{\not\perp}_{\mathsf{trans}}$ from D_{sim} with probability at least $\epsilon^2/(C \cdot \mathsf{Q}_{\mathsf{H\text{-}P}})$ for some universal constant $C > 0$. However, by Lemma 4.5, we must have $\epsilon^2/(C \cdot \mathsf{Q}_{\mathsf{H\text{-}P}}) = \mathsf{negl}(\lambda)$, which establishes $\epsilon = \mathsf{negl}(\lambda)$ as desired. □

Multi-Proof Extractability.

Theorem 5.8. *The* NIZK $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ *in Figs. 4 and 5 is quantumly multi-proof extractable with* $(c_1, e_1, e_2) = (1, 2, 1)$ *and* $p(\lambda) = \mathsf{poly}(\lambda)$ *if the* $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{d}'},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}'},Q}$, *and* $\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}$ *problems are hard.*

Proof. Due to the way we organized the proof of Theorem 4.6 in the classical ROM, we are able to reuse most parts of the proof. To start, the proof of CRS indistinguishability is identical to those provided in Theorem 4.6.

Regarding the proof of straight-line extractability, the only part that requires a tailored argument for QROM security is in the proof of Lemma 4.7, which established that if a *classical* PPT adversary $\mathcal{A}$ outputs a valid proof, then there must have been multiple challenges for which it could have succeeded on. For a *quantum* polynomial time adversary $\mathcal{A}$, we must take into account that it can make quantum random oracle queries to the hash function. Otherwise, the proof of Lemma 4.10 and the rest of the proof of straight-line extractability remains intact since they are purely statistical arguments which hold regardless of being in the classical or quantum ROM.

Our goal is to thus to modify the proof of Lemma 4.7 to the following Lemma 5.9 so that the claim holds even against QPT adversaries. We note that since the lower bound on Γ_2 is altered, the runtime of our straight-line extractor $\mathsf{Multi\text{-}Extract}$ will be proportional to $O(\frac{\mathsf{Q}_{\mathsf{S}} \cdot \mathsf{Q}_{\mathsf{H}}^2}{\mu})$ rather than $O(\frac{\mathsf{Q}_{\mathsf{H}}}{\mu})$ as in the classical setting.

Lemma 5.9. *Let us define the transcript* $(a_1, \mathbf{c}_1, a_2, \mathbf{c}_2, \mathsf{resp})$ *and sets* $(\Gamma_{1,i})_{i\in[\tau]}$ *and* Γ_2 *as in Lemma 4.7. Then, for any* $\mathsf{Q}_{\mathsf{H}} = \mathsf{poly}(\lambda)$ *and QPT adversary* $\mathcal{A}$ *that makes at most* Q_{H} *(quantum) random oracle queries with*

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

we have,

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \tau) \xleftarrow{\$} \mathcal{S}_{\mathsf{crs}}(1^\lambda),\\ \{(\mathsf{X}_k, \pi^{\mathsf{m}}_k)\}_{k\in[\mathsf{Q}_{\mathsf{S}}]} \xleftarrow{\$} \mathcal{A}^{|\mathsf{H}_{\mathsf{m}}\rangle}(1^\lambda, \widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}),\end{array} : \begin{array}{c}\forall k \in [\mathsf{Q}_{\mathsf{S}}], \mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}_k, \pi^{\mathsf{m}}_k) = \top\\ \wedge\ \exists i \in [\tau], |\Gamma_{1,i}(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq 3\\ \wedge\ |\Gamma_2(\mathsf{X}_k, a_{1,k}, \mathbf{c}_{1,k}, a_{2,k}, \mathbf{c}_{2,k})| \geq \frac{\mu}{16\cdot\mathsf{Q}_{\mathsf{S}}\cdot(\mathsf{Q}_{\mathsf{H}}+1)^2} \cdot |C_{\mathsf{ham}}|\end{array}\right] \geq \frac{1}{2}\cdot\mu(\lambda) - \mathsf{negl}(\lambda).$$

Proof. The only difference between the proof in the classical and quantum setting is how we upper bound Corollaries 4.8 and 4.9 in Lemma 4.7. For notational simplicity, we 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 $(k, i) \in [\mathsf{Q}_{\mathsf{S}}] \times [\tau]$. Let T_2 be a positive integer, which we define shortly after. We denote by $\mathsf{ValidProofs}$ the event that $\mathsf{Verify}^{\mathsf{H}_{\mathsf{m}}}(\widetilde{\mathsf{crs}}^{\mathsf{m}}_{\mathsf{NIZK}}, \mathsf{X}_k, \pi^{\mathsf{m}}_k) = \top$ for all $k \in [\mathsf{Q}_{\mathsf{S}}]$, and when the context is clear, we omit the sampling probability space. Then, we can rewrite $\mathcal{A}$'s advantage as follows:

$$\begin{aligned}
\mu &\leq \Pr[\mathsf{ValidProofs}]\\
&= \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_{\mathsf{S}}], \left(\exists i \in [\tau], \left|\Gamma^{(k)}_{1,i}\right| \geq 3\right) \wedge \left(\left|\Gamma^{(k)}_2\right| \geq T_2\right)\right)\right]\\
&\quad + \Pr\left[\mathsf{ValidProofs} \wedge \left(\exists k \in [\mathsf{Q}_{\mathsf{S}}], \left(\forall i \in [\tau], \left|\Gamma^{(k)}_{1,i}\right| < 3\right) \vee \left(\left|\Gamma^{(k)}_2\right| < T_2\right)\right)\right]\\
&\leq \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q}_{\mathsf{S}}], \left(\exists i \in [\tau], \left|\Gamma^{(k)}_{1,i}\right| \geq 3\right) \wedge \left(\left|\Gamma^{(k)}_2\right| \geq T_2\right)\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{k\in[\mathsf{Q_S}]} \Pr\left[\mathsf{ValidProofs} \wedge \forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right] && (17)\\
&+ \sum_{k\in[\mathsf{Q_S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left|\Gamma_{2}^{(k)}\right| < T_2\right] && (18)\\
&\le \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q_S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_{2}^{(k)}\right| \ge T_2\right)\right)\right]\\
&\quad + 64 \cdot \mathsf{Q_S} \cdot (\mathsf{Q_H}+1)^4 \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau} + 8 \cdot \mathsf{Q_S} \cdot (\mathsf{Q_H}+1)^2 \cdot \frac{T_2}{|C_{\mathsf{ham}}|},
\end{aligned}
$$

where the second inequality follows from the union bound, and the third inequality is due to Corollaries 5.10 and 5.11 that establish upper bounds on Eqs. (17) and (18), respectively. We first finish the proof of Lemma 5.9.

By plugging in $T_2 := \frac{\mu}{16\cdot\mathsf{Q_S}\cdot(\mathsf{Q_H}+1)^2} \cdot |C_{\mathsf{ham}}|$ in the above inequality, we obtain the following

$$
\Pr\left[\mathsf{ValidProofs} \wedge \left(\forall k \in [\mathsf{Q_S}], \left(\exists i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| \ge 3\right) \wedge \left(\left|\Gamma_{2}^{(k)}\right| \ge T_2\right)\right)\right] \ge \frac{\mu}{2} - 64 \cdot \mathsf{Q_S} \cdot (\mathsf{Q_H}+1)^4 \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau}.
$$

Due to our parameter setting (i.e., $\frac{q'}{2} \approx (2d)^{\tau'}$ and $1/(2d)^{\tau\cdot\tau'} = \mathsf{negl}(\lambda)$), for any $\mathsf{Q_S} = \mathsf{poly}(\lambda)$ and $\mathsf{Q_H} = \mathsf{poly}(\lambda)$, the term being subtracted from $\frac{\mu}{2}$ is negligible. Thus we obtain the desired bound.

It remains to prove the following Corollaries 5.10 and 5.11.

Corollary 5.10. *We have* $\sum_{k\in[\mathsf{Q_S}]} \Pr\left[\mathsf{ValidProofs} \wedge \forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right] \le 64\cdot\mathsf{Q_S}\cdot(\mathsf{Q_H}+1)^4 \cdot \left(\frac{2}{q'} + \frac{1}{(2d)^{\tau'}}\right)^{\tau}$.

Proof. We further modify the equation as follows,

$$
\begin{aligned}
&\sum_{k\in[\mathsf{Q_S}]} \Pr\left[\mathsf{ValidProofs} \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\right]\\
&= \sum_{k\in[\mathsf{Q_S}]} \sum_{J_k\in[0:\tau]} \sum_{\substack{S\subseteq[\tau]\\ \text{s.t. } |S|=J_k}} \Pr\left[\begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array} \wedge \begin{pmatrix}\forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)}\\ \forall i \in [\tau]\backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)}\end{pmatrix}\right]\\
&\le \sum_{k\in[\mathsf{Q_S}]} \sum_{J_k\in[0:\tau]} \sum_{\substack{S\subseteq[\tau]\\ \text{s.t. } |S|=J_k}} \Pr\left[\begin{pmatrix}\forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)}\\ \forall i \in [\tau]\backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)}\end{pmatrix} \middle| \begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right]\\
&\le \sum_{k\in[\mathsf{Q_S}]} \sum_{J_k\in[0:\tau]} \sum_{\substack{S\subseteq[\tau]\\ \text{s.t. } |S|=J_k}} \Pr\left[\left(\forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right]\\
&\qquad\qquad \cdot \Pr\left[\left(\forall i \in [\tau]\backslash S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right] \qquad (19)
\end{aligned}
$$

where $c_{1,k,i}$ is the i-th element in the k-th second-flow challenge $\mathbf{c}_{1,k}$ included in π_k^{m} output by adversary $\mathcal{A}$. The first inequality follows from taking the conditional probability and the second inequality follows from the fact that the output of the random oracle is uniform and thus the distributions of each $(c_{1,k,i})_{i\in[\tau]}$ are independent (even though $\mathcal{A}$ can freely chose which $(c_{1,k,i})_{i\in[\tau]}$ to output). In other words, for each $k \in [\mathsf{Q_S}]$ and $\mathbf{c}_{1,k} = (c_{1,k,i})_{i\in[\tau]}$, $c_{1,k,i}$ is either in $\Gamma_{1,i}^{(k)}$ of size at most 2 or not, and J_k counts the number of $c_{1,k,i} \in \Gamma_{1,i}^{(k)}$ in $\mathbf{c}_{i,k}$.

We use Lemma 2.27 to bound Eq. (19). That is, given a (possibly unbounded) quantum adversary $\mathcal{A}$, we construct quantum adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ against the generic search problem with bounded probabilities.

Constructing $\mathcal{B}_1$. Let us fix (k, J_k, S) in the summand. We first bound the probability that $c_{1,k,i} \in \Gamma_{1,i}^{(k)}$ for all $i \in S$. We assume the domain D of the function G, which $\mathcal{B}_1$ will be given oracle access to, to be the same as that of H_{m}. $\mathcal{B}_1$ then prepares the set of reals $(\lambda_z)_{z\in D}$ as follows: if $z = (\mathsf{X}, 1, a_1)$, then define

$$\lambda_z := \frac{\prod_{i\in S} \left|\Gamma_{1,i}^k\right|}{\mathbb{Z}_{q'}^{J_k}} \leq \left(\frac{2}{q'}\right)^{J_k},$$

otherwise, $\lambda_z := 0$, where we use the fact $\left|\Gamma_{1,i}^k\right| \leq 2$ for all $i \in S$. It then outputs $(\lambda_z)_{z\in D}$ to the challenger. By setting $\lambda := \left(\frac{2}{q'}\right)^{J_k}$, it is clear that this is a valid input for the generic search problem. Define the sets $I := \otimes_{i\in[\tau]}\mathbb{Z}_{q'}$ and $I_{\mathsf{bad}} := (\otimes_{i\in S}\Gamma_{i,1}^{(k)}) \otimes (\otimes_{i\in[\tau]\setminus S}\mathbb{Z}_{q'})$, where we assume the latter is properly reordered with respect to $i \in [\tau]$. $\mathcal{B}_1$ then samples random functions RF_1, RF_2, and RF_3 with domain D and range the same as H_{m} conditioned on $\mathsf{RF}_1(z) \in I\setminus I_{\mathsf{bad}}$ and $\mathsf{RF}_2(z) \in I_{\mathsf{bad}}$ for all inputs z of the form $(\mathsf{X}, 1, a_1)$. Finally, $\mathcal{B}_1$ simulates $\mathcal{A}$ by using its oracle G. Specifically, to simulate an oracle query to $\mathsf{H}_{\mathsf{m}}(z)$, if z is not of the form $(\mathsf{X}, 1, a_1)$, then it returns $\mathsf{RF}_3(z)$. Otherwise, it returns $\mathsf{RF}_1(z)$ if $0 \leftarrow \mathsf{G}(z)$ and returns $\mathsf{RF}_2(z)$ if $1 \leftarrow \mathsf{G}(z)$. Here, note that $\mathcal{B}_1$ can perform this computation on superpositions $\sum_z \alpha_z |z\rangle$, where α_z is the amplitude. When $\mathcal{A}$ outputs a proof π_k^{m}, $\mathcal{B}_1$ extracts (X_k, a_k) and then outputs $z = (\mathsf{X}_k, 1, a_k)$.

Let us analyze $\mathcal{B}_1$. First of all, it can be checked that $\mathcal{B}_1$ simulates the view to $\mathcal{A}$ perfectly since the output distribution of H_{m} is perfectly simulated using G. Moreover, if $\mathcal{A}$ succeeds in outputting a valid proof π_k^{m} such that $(c_i, k, i) \in \Gamma_{1,i}^{(k)}$ for all $i \in S$, then the z that $\mathcal{B}_1$ extracts must satisfy $\mathsf{H}_{\mathsf{m}}(z) = \mathsf{RF}_2(z) \in I_{\mathsf{bad}}$. Therefore, by definition $\mathsf{G}(z) = 1$, and in particular, the success probability of $\mathcal{B}_1$ is the same as $\mathcal{A}$. Then, assuming the hardness of the generic search problem with bounded probabilities, we must have

$$\Pr\left[\left(\forall i \in S, c_{1,k,i} \in \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right] \leq 8 \cdot (\mathsf{Q}_{\mathsf{H}} + 1)^2 \cdot \lambda = 8 \cdot (\mathsf{Q}_{\mathsf{H}} + 1)^2 \cdot \left(\frac{2}{q'}\right)^{J_k}.$$

Constructing $\mathcal{B}_2$. We next bound the probability that $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\setminus S$. By definition of $\Gamma_{1,i}^{(k)}$, if $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$, then there is only one set of $\boldsymbol{\beta}_i := (\beta_{k,i,j})_{j\in[\tau']} \in C_X^{\tau'}$ that can be included in a valid transcript containing $c_{1,k,i}$. That is, if $\mathsf{trans} := (a_1, \mathbf{c}_1 = (c_i)_{i\in[\tau]}, a_2, \mathbf{c}_2 = (\boldsymbol{\beta}, \beta'), \mathsf{resp})$ is a valid transcript, then $\boldsymbol{\beta}_i$ is guaranteed to be included in $\boldsymbol{\beta}$. Let $(\boldsymbol{\beta}_i)_{i\in[\tau]\setminus S} \subseteq C_X^{(\tau-J_k)\cdot\tau'}$ denote those unique challenges corresponding to $\{c_{1,k,i}\}_{i\in[\tau]\setminus S}$. We are now ready to describe $\mathcal{B}_2$ against the generic search problem with bounded probabilities. Let us fix (k, J_k, S). We assume the domain D of the function G, which $\mathcal{B}_2$ will be given oracle access to, to be the same as that of H_{m}. $\mathcal{B}_2$ then prepares the set of reals $(\lambda'_z)_{z\in D}$ as follows: if $z = (\mathsf{X}, 1, a_1, \mathbf{c}_1, a_2)$ and $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\setminus S$, then define $\lambda'_z := \left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k}$, and otherwise, $\lambda'_z := 0$, where recall $|C_X| = 2d$. It then outputs $(\lambda'_z)_{z\in D}$ to the challenger. Define the sets $I' := \left(\otimes_{i\in[\tau]}\left(\otimes_{j\in[\tau']}C_X\right)\right)$ and $I'_{\mathsf{bad}} := \left(\otimes_{i\in S}\left(\otimes_{j\in[\tau']}C_X\right)\right) \otimes \left(\otimes_{i\in[\tau]\setminus S}\{\boldsymbol{\beta}_i\}\right)$, where we assume the latter is properly reordered with respect to $i \in [\tau]$. $\mathcal{B}_2$ then samples random functions RF_1, RF_2, and RF_3 with domain D and range the same as H_{m} conditioned on $\mathsf{RF}_1(z) \in I'\setminus I'_{\mathsf{bad}}$ and $\mathsf{RF}_2(z) \in I'_{\mathsf{bad}}$ for all inputs z of the form $(\mathsf{X}, 1, a_1, \mathbf{c}_1, a_2)$ such that $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\setminus S$. Finally, $\mathcal{B}_2$ simulates $\mathcal{A}$ by using its oracle G. Specifically, to simulate an oracle query to $\mathsf{H}(z)$, if z is not of the form $(\mathsf{X}, 1, a_1, \mathbf{c}_1, a_2)$ and $c_{1,k,i} \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\setminus S$, then it returns $\mathsf{RF}_3(z)$. Otherwise, it returns $\mathsf{RF}_1(z)$ if $0 \leftarrow \mathsf{G}(z)$ and returns $\mathsf{RF}_2(z)$ if $1 \leftarrow \mathsf{G}(z)$. Here, note that $\mathcal{B}_2$ can perform this computation on superpositions $\sum_z \alpha_z |z\rangle$, where α_z is the amplitude. When $\mathcal{A}$ outputs a proof π_k^{m}, $\mathcal{B}_1$ extracts $(\mathsf{X}_k, a_k, \mathbf{c}_{1,k}, a_2)$ and then outputs $z = (\mathsf{X}_k, 1, a_k, \mathbf{c}_{1,k}, a_2)$.

Let us analyze $\mathcal{B}_2$. First of all, it can be checked that $\mathcal{B}_2$ simulates the view to $\mathcal{A}$ perfectly since the output distribution of H_{m} is perfectly simulated using G. Moreover, if $\mathcal{A}$ succeeds in outputting a valid proof π_k^{m} such that $(c_i, k, i) \notin \Gamma_{1,i}^{(k)}$ for all $i \in [\tau]\setminus S$, then the z that $\mathcal{B}_1$ extracts must satisfy $\mathsf{H}_{\mathsf{m}}(z) = \mathsf{RF}_2(z) \in I'_{\mathsf{bad}}$. Therefore, by definition $\mathsf{G}(z) = 1$, and in particular, the success probability of $\mathcal{B}_2$ is the same as $\mathcal{A}$. Then, assuming the hardness of the generic search problem with bounded probabilities, we must have

$$\Pr\left[\left(\forall i \in [\tau]\setminus S, c_{1,k,i} \notin \Gamma_{1,i}^{(k)}\right) \middle| \begin{array}{c}\mathsf{ValidProofs}\\ \wedge \left(\forall i \in [\tau], \left|\Gamma_{1,i}^{(k)}\right| < 3\right)\end{array}\right] \leq 8 \cdot (\mathsf{Q}_{\mathsf{H}} + 1)^2 \cdot \lambda' = 8 \cdot (\mathsf{Q}_{\mathsf{H}} + 1)^2 \cdot \left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k}.$$

Combining the two arguments, we upper bound Eq. (19) as follow:

$$\begin{aligned}
&\sum_{k\in[\mathsf{Q_S}]}\sum_{J_k\in[0:\tau]}\sum_{\substack{S\subseteq[\tau]\\ \text{s.t. } |S|=J_k}} 64\cdot(\mathsf{Q_H}+1)^4\cdot\left(\frac{2}{q'}\right)^{J_k}\cdot\left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k}\\
=&\sum_{k\in[\mathsf{Q_S}]} 64\cdot(\mathsf{Q_H}+1)^4\left(\sum_{J_k\in[0:\tau]}\binom{\tau}{J_k}\left(\frac{2}{q'}\right)^{J_k}\cdot\left(\frac{1}{(2d)^{\tau'}}\right)^{\tau-J_k}\right)\\
=&\sum_{k\in[\mathsf{Q_S}]} 64\cdot(\mathsf{Q_H}+1)^4\cdot\left(\frac{2}{q'}+\frac{1}{(2d)^{\tau'}}\right)^{\tau}\\
\leq&64\cdot\mathsf{Q_S}\cdot(\mathsf{Q_H}+1)^4\cdot\left(\frac{2}{q'}+\frac{1}{(2d)^{\tau'}}\right)^{\tau},
\end{aligned}$$

where the second equality follows from the binomial expansion. This completes the proof. □

Corollary 5.11. *We have* $\sum_{k\in[\mathsf{Q_S}]}\Pr\left[\mathsf{ValidProofs}\wedge\left|\Gamma_2^{(k)}\right|<T_2\right]\leq 8\cdot\mathsf{Q_S}\cdot(\mathsf{Q_H}+1)^2\cdot\frac{T_2}{|C_{\mathsf{ham}}|}$.

Proof. Similarly to the proof of Corollary 5.10, we can use the generic search problem with bounded probabilities to clam the following:

$$\sum_{k\in[\mathsf{Q_S}]}\Pr\left[\mathsf{ValidProofs}\wedge\left|\Gamma_2^{(k)}\right|<T_2\right]\leq\sum_{k\in[\mathsf{Q_S}]}8\cdot(\mathsf{Q_H}+1)^2\cdot\frac{T_2}{|C_{\mathsf{ham}}|}=8\cdot\mathsf{Q_S}\cdot(\mathsf{Q_H}+1)^2\cdot\frac{T_2}{|C_{\mathsf{ham}}|}.$$

□

□

□

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

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May / June 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, Heidelberg, August 2010.

[Abe01] Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, Heidelberg, May 2001.

[AEB20a] Nabil Alkeilani Alkadri, Rachid El Bansarkhani, and Johannes Buchmann. BLAZE: Practical lattice-based blind signatures for privacy-preserving applications. In Joseph Bonneau and Nadia Heninger, editors, *FC 2020*, volume 12059 of *LNCS*, pages 484–502. Springer, Heidelberg, February 2020.

[AEB20b] Nabil Alkeilani Alkadri, Rachid El Bansarkhani, and Johannes Buchmann. On lattice-based interactive protocols: An approach with less or no aborts. In Joseph K. Liu and Hui Cui, editors, *ACISP 20*, volume 12248 of *LNCS*, pages 41–61. Springer, Heidelberg, November / December 2020.

[AHJ21] Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. Blindor: An efficient lattice-based blind signature scheme from or-proofs. In *CANS*, pages 95–115. Springer, 2021.

[AKSY21a] Shweta Agrawal, Elena Kirshanova, Damien Stehle, and Anshu Yadav. Can round-optimal lattice-based blind signatures be practical? *Cryptology ePrint Archive*, 2021.

[AKSY21b] Shweta Agrawal, Elena Kirshanova, Damien Stehle, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. Cryptology ePrint Archive, Paper 2021/1565, 2021. https://eprint.iacr.org/2021/1565.

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 470–499. Springer, Heidelberg, August 2020.

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, Heidelberg, August 2000.

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014.

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 551–572. Springer, Heidelberg, December 2014.

[BDK+21] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. *To Appear at EUROCRYPT*, 2021.

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Heidelberg, September 2018.

[BFW15] David Bernhard, Marc Fischlin, and Bogdan Warinschi. Adaptive proofs of knowledge in the random oracle model. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 629–649. Springer, Heidelberg, March / April 2015.

[BG14] Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, Heidelberg, February 2014.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. On the security of one-witness blind signature schemes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 82–99. Springer, Heidelberg, December 2013.

[BLL+21] Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Heidelberg, October 2021.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Heidelberg, August 2019.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.

[Boy10] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer, Heidelberg, May 2010.

[Bra94] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Heidelberg, August 1994.

[Cam97] Jan Camenisch. Efficient and generalized group signatures. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 465–479. Springer, Heidelberg, May 1997.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, Heidelberg, August 1990.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

[Cha88] David Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 177–182. Springer, Heidelberg, May 1988.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Heidelberg, May / June 2010.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, October 2012.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001.

[CPS+20] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. ModFalcon: Compact signatures based on module-NTRU lattices. In Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese, editors, *ASIACCS 20*, pages 853–866. ACM Press, October 2020.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Heidelberg, August 2020.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Heidelberg, August 2019.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018. https://tches.iacr.org/index.php/TCHES/article/view/839.

[dLS18] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 574–591. ACM Press, October 2018.

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, December 2020.

[ESLR22] Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. *Cryptology ePrint Archive*, 2022.

[FHS15] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, August 2015.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Heidelberg, August 2006.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT*, pages 244–251. Springer, 1992.

[FPS20] Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Heidelberg, May 2020.

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, Heidelberg, May / June 2010.

[Goo22] Vpn by google one, explained. https://one.google.com/about/vpn/howitworks, 2022. Accessed: 2022-02-02.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GRS+11] Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648. Springer, Heidelberg, August 2011.

[HKLN20] Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 500–529. Springer, Heidelberg, August 2020.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.

[Kat21] Shuichi Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 580–610, Virtual Event, August 2021. Springer, Heidelberg.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, Heidelberg, April / May 2018.

[KLX20] Julia Kastner, Julian Loss, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. *To Appear at PKC*, 2020.

[KNYY21] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Round-optimal blind signatures in the plain model from classical and quantum standard assumptions. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 404–434. Springer, Heidelberg, October 2021.

[Lin08] Yehuda Lindell. Lower bounds and impossibility results for concurrent self composition. *Journal of Cryptology*, 21(2):200–249, April 2008.

[LNP22a] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Efficient lattice-based blind signatures via gaussian one-time signatures. *To Appear at PKC*, 2022.

[LNP22b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *To Appear at Crypto*, 2022.

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1051–1070. ACM Press, November 2020.

[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 215–241. Springer, Heidelberg, May 2021.

[LSK+19] Huy Quoc Le, Willy Susilo, Thanh Xuan Khuc, Minh Kim Bui, and Dung Hoang Duong. A blind signature from module latices. In *Dependable and Secure Computing (DSC)*, pages 1–8. IEEE, 2019.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 1219–1234. ACM Press, May 2012.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 326–355. Springer, Heidelberg, August 2019.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[OO92] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 324–337. Springer, Heidelberg, August 1992.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 109–118. ACM Press, June 2011.

[Pei10] Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Heidelberg, August 2010.

[PFH+18] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. Technical report, 2018. Available at https://falcon-sign.info/.

[PHBS19] D. Papachristoudis, D. Hristu-Varsakelis, F. Baldimtsi, and G. Stephanides. Leakage-resilient lattice-based partially blind signatures. Cryptology ePrint Archive, Report 2019/1452, 2019. https://eprint.iacr.org/2019/1452.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

[Rüc10] Markus Rückert. Lattice-based blind signatures. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 413–430. Springer, Heidelberg, December 2010.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

[Sch01] Claus-Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Heidelberg, November 2001.

[SG98] Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 1–16. Springer, Heidelberg, May / June 1998.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018.

[TZ22] Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 782–811. Springer, Heidelberg, May / June 2022.

[YL19] Xun Yi and Kwok-Yan Lam. A new blind ECDSA scheme for bitcoin transaction anonymity. In Steven D. Galbraith, Giovanni Russello, Willy Susilo, Dieter Gollmann, Engin Kirda, and Zhenkai Liang, editors, *ASIACCS 19*, pages 613–620. ACM Press, July 2019.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7-8):557–567, 2015.

A Omitted Preliminaries

A.1 Proof Sketch of Modified Trapdoor Sampling

Let us rewrite $\mathbf{a} \in R_q^{k'}$ as $[\mathbf{a}_1|\mathbf{a}_2]$, where $\mathbf{a}_1 \in R_q^{k_1}$ and $\mathbf{a}_2 \in R_q^{k_2}$. Let $\mathbf{I} \in R_q^{k' \times k'}$ denote the identity matrix. Then observe we have

$$[\mathbf{a}|\mathbf{a}\mathbf{R}^* + t \cdot \mathbf{g}] \begin{bmatrix} -c \cdot \mathbf{R} \\ -\mathbf{R}' \\ c \cdot \mathbf{I} \end{bmatrix} = \left[\mathbf{a}_1 \middle| \mathbf{a}_2 \middle| [\mathbf{a}_1|\mathbf{a}_2] \begin{bmatrix} \mathbf{R} \\ \frac{1}{c} \cdot \mathbf{R}' \end{bmatrix} + t \cdot \mathbf{g}\right] \begin{bmatrix} -c\mathbf{R} \\ -\mathbf{R}' \\ c \cdot \mathbf{I} \end{bmatrix} = c \cdot t \cdot \mathbf{g}.$$

This has the desired form required to run the almost identical sampling algorithm provided in [MP12, Section 5.4]. At a high level, given any $u \in R_q$, we can sample a short vector $\mathbf{z} \in R^{k'}$ such that $\mathbf{g}\mathbf{z}^\top = (c \cdot t)^{-1} \cdot u$ by using the public trapdoor $\mathbf{T}_\mathbf{g}$ of $\mathbf{g}$, where recall c and t are invertible over R_q. Then, $\mathbf{e}' = \begin{bmatrix} -c \cdot \mathbf{R} \\ -\mathbf{R}' \\ c \cdot \mathbf{I} \end{bmatrix} \mathbf{z}^\top$ is short and satisfies $[\mathbf{a}|\mathbf{a}\mathbf{R}^* + t \cdot \mathbf{g}]\mathbf{e}'^\top = u$. However, since $\mathbf{e}'$ does not yet have a spherical Gaussian distribution, we cannot output this. [MP12] shows how to correct this idea by using the convolution technique from [Pei10]. It can be checked that all the arguments made in [MP12] holds for our case, where the only difference is that our Gaussian parameter increases by a factor of c.

A.2 Forking Lemma

The forking lemma was originally introduced by Pointcheval and Stern [PS00] in the context of signature schemes. The lemma was later reformulated by Bellare and Neven [BN06] which extracts the purely probabilistic nature of the forking lemma.

Lemma A.1 (Forking Lemma). *Fix an integer $q \geq 1$ and a set $\mathcal{H}$ of size $h \geq 2$. Let $\mathcal{A}$ be a randomized algorithm, where on input $\mathsf{par}, h_1, \cdots, h_q$, algorithm $\mathcal{A}$ returns a pair; the first element is an integer in the range $(0, \cdots, q)$ and the second element σ is what we refer to as a* side output*. Let IG be a randomized algorithm called the* input generator*. The accepting probability of $\mathcal{A}$, denoted acc, is defined below:*

$$\mathsf{acc} = \Pr[\mathsf{par} \xleftarrow{\$} \mathsf{IG},\ (h_1, \cdots, h_q) \xleftarrow{\$} \mathcal{H}^q,\ (J, \sigma) \xleftarrow{\$} \mathcal{A}(\mathsf{par}, h_1, \cdots, h_q) : J \geq 1].$$

The forking algorithm $\mathsf{Fork}_\mathcal{A}$ *associated to $\mathcal{A}$ is a randomized algorithm that takes input* par *and proceeds as in Fig. 10, where ϵ_1 and ϵ_2 are arbitrary strings. Let*

$$\mathsf{frk} = \Pr[\mathsf{par} \xleftarrow{\$} \mathsf{IG};\ (b, (\sigma_1, \sigma_2)) \xleftarrow{\$} \mathsf{Fork}_\mathcal{A}(\mathsf{par})\ :\ b = 1].$$

Then,

$$\mathsf{frk} \geq \mathsf{acc} \cdot \left(\frac{\mathsf{acc}}{q} - \frac{1}{h}\right). \tag{20}$$

Algorithm $\mathsf{Fork}_{\mathcal{A}}(\mathsf{par})$
Pick coin ρ for $\mathcal{A}$ at random.
$(h_1, \cdots, h_q) \xleftarrow{\$} \mathcal{H}^q$
$(I, \sigma) := \mathcal{A}(\mathsf{par}, h_1, \cdots, h_q; \rho)$
If $I = 0$ **then**
return $(0, (\bot, \bot))$
$(h'_I, \cdots, h'_q) \xleftarrow{\$} \mathcal{H}^{q-I+1}$
$(I', \sigma') := \mathcal{A}(\mathsf{par}, h_1, \cdots, h_{I-1}, h'_I, \cdots h'_q; \rho)$
If $I = I' \ \wedge \ h_I \neq h'_I$ **then**
return $(1, (\sigma_1, \sigma_2))$
Else
return $(0, (\bot, \bot))$

Figure 10: Description of the forking algorithm $\mathsf{Fork}_{\mathcal{A}}$.

A.3 Partially Blind Signature

We provide the definition of partially blind signatures [AO00]. For simplicity, we give a definition focusing on round-optimal (i.e., two-round) partially blind signatures.

Definition A.2 (Partially Blind Signature). *A round-optimal partially blind signature scheme* Π_{BS} *with message space* $\mathcal{M}$ *and common message space* $\mathcal{M}_c$ *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*.*

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

$\mathcal{S}_2(\mathsf{sk}, \gamma, \rho_1) \to \rho_2$**:** *This is the signer's second message generation algorithm that takes as input a signing key* sk*, a common message* $\gamma \in \mathcal{M}_c$*, 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*.*

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

Definition A.3 (Correctness). *A partially blind signature is* correct *if for any* $\lambda \in \mathbb{N}$*,* $\gamma \in \mathcal{M}_c$*, and* $\mathsf{M} \in \mathcal{M}$*, we have*

$$\Pr\left[\begin{array}{l} (\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{BSGen}(1^\lambda) \\ (\rho_1, \mathsf{st}_{\mathcal{U}}) \xleftarrow{\$} \mathcal{U}_1(\mathsf{vk}, \gamma, \mathsf{M}) \\ \rho_2 \xleftarrow{\$} \mathcal{S}_2(\mathsf{sk}, \gamma, \rho_1) \\ \Sigma \xleftarrow{\$} \mathcal{U}_{\mathsf{der}}(\mathsf{st}_{\mathcal{U}}, \rho_2) \end{array} : \mathsf{BSVerify}(\mathsf{vk}, \gamma, \mathsf{M}, \Sigma) = \top\right] = 1 - \mathsf{negl}(\lambda).$$

Definition A.4 (One-More Unforgeability). *A partially blind signature is* one-more unforgeable *if for any* $\mathsf{Q} = \mathsf{poly}(\lambda)$ *and QPT adversary* $\mathcal{A}$ *that for each common message* γ*, it makes at most* Q *(classical) queries containing the same* γ *to the singer oracle, we have*

$$\mathsf{Adv}^{\mathsf{OMU}}_{\Pi_{\mathsf{BS}}}(\mathcal{A}) := \Pr\left[\begin{array}{l} (\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{BSGen}(1^\lambda) \\ (\gamma, \{(\mathsf{M}_i, \Sigma_i)\}_{i \in [\mathsf{Q}+1]}) \xleftarrow{\$} \mathcal{A}^{\mathcal{S}_2(\mathsf{sk}, \cdot, \cdot)}(\mathsf{vk}) \end{array}\right.$$

$$: \begin{array}{l} \mathsf{BSVerify}(\mathsf{vk}, \gamma, \mathsf{M}_i, \Sigma_i) = \top \text{ for all } i \in [\mathsf{Q}+1] \\ \wedge \{\mathsf{M}_i\}_{i\in[\mathsf{Q}+1]} \text{ is pairwise distinct} \end{array} \Bigg] = \mathsf{negl}(\lambda)$$

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 A.5 (Partial Blindness Under Malicious Keys). *To define partial 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*, a common message* γ*, and a pair of messages* $(\mathsf{M}_0, \mathsf{M}_1)$ *to the challenger.*

First Message. *The challenger generates* $(\rho_{1,b}, \mathsf{st}_{\mathcal{U},b}) \xleftarrow{\$} \mathcal{U}_1(\mathsf{vk}, \gamma, \mathsf{M}_b)$ *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* $\mathsf{BSVerify}(\mathsf{vk}, \gamma, \mathsf{M}_b, \Sigma_b) = \bot$ *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 partially blind signature is* partially blind against malicious senders *if for any QPT adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathsf{blind}}_{\Pi_{\mathsf{BS}}}(\mathcal{A}) := \left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right| = \mathsf{negl}(\lambda).$$

B Tools to Argue Single-Proof Extractability of NIZKs in the QROM

In this section, we provide known techniques to argue single-proof extractability of NIZKs in the QROM. Unlike the classical case, we cannot simply rewind the cheating prover to extract the witness since it may disrupt the quantum prover's internal state. Specifically, we cannot rely on the standard argument to lower bound the probability that the prover also succeeds after being rewound with non-negligible advantage. The contents of this section are prior results but we decided to create a new section rather than including it in Appendix A for better readability.

B.1 Sigma Protocol

We recall the definition of sigma protocol. In the following, we consider the statement X to be generated by some instance generator IGen. We note that a sigma protocol defined with respect to an instance generator IGen can be thought of as an identification protocol (with slightly different security definitions).

Definition B.1 (Sigma-Protocol). *A sigma-protocol* Π_Σ *for relations* $(\mathcal{R}, \mathcal{R}_{\mathsf{gap}})$ *is defined by a tuple of algorithms* $(\mathsf{Prove} = (\mathsf{Prove}_1, \mathsf{Prove}_2), \mathsf{Verify})$*, where* Verify *is a deterministic polynomial time algorithm. We assume the relation* $\mathcal{R}$ *defines the set of all commitments* ComSet*, challenges* ChSet*, and responses* ResSet*. A sigma-protocol proceeds as follows:*

1. *The prover, on input* $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$*, runs* $(\alpha, \mathsf{st}) \xleftarrow{\$} \mathsf{Prove}_1(\mathsf{X}, \mathsf{W})$ *and returns* $\alpha \in \mathsf{ComSet}$ *to the verifier;*
2. *The verifier then samples a challenge* $\beta \xleftarrow{\$} \mathsf{ChSet}$ *and returns it to the prover;*
3. *The prover sends a response* $\gamma \xleftarrow{\$} \mathsf{Prove}_2(\mathsf{X}, \mathsf{W}, (\alpha, \beta, \mathsf{st}))$ *to the verifier, where* $\gamma \in \mathsf{ResSet} \cup \{\bot\}$ *and* $\bot \notin \mathsf{ResSet}$ *is a special symbol indicating failure. Finally, the verifier runs* $\mathsf{Verify}(\mathsf{X}, (\alpha, \beta, \gamma))$ *and outputs* $\top$ *for acceptance and* $\bot$ *for rejection.*

The transcript (α, β, γ) *is called a* valid transcript *if* $\mathsf{Verify}(\mathsf{X}, (\alpha, \beta, \gamma)) = \top$. *Finally, we define an* instance generator IGen *such that on input the security parameter* 1^λ*, it outputs a pair* $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$.

We typically require a sigma-protocol to satisfy correctness, (non-abort) honest-verifier zero-knowledge, and special soundness. Below we only define special soundness since the implicit sigma protocols appearing in our NIZK constructions are indirectly proven to satisfy correctness and (non-abort) honest-verifier zero-knowledge.

Definition B.2 (Relaxed Two-Special Soundness). *A sigma-protocol* Π_Σ *has* relaxed two-special soundness *if there is a deterministic PT algorithm* $\mathsf{Extract}_{\mathsf{ss}}$ *such that given any two valid transcripts* $(\alpha, \{(\beta_i, \gamma_i)\}_{i\in[2]})$ *for any statement* $\mathsf{X} \in \mathcal{L}_\mathcal{R}$ *with* $\beta_1 \neq \beta_2$*, it outputs a witness* W *such that* $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}_{\mathsf{gap}}$.

We can use the Fiat-Shamir transform to make a sigma protocol non-interactive. Formally, the prover generates the challenge β by $\mathsf{H}(\mathsf{X}, \alpha)$ and finishes the sigma protocol on its own. Classically, we know that if the if the underlying sigma protocol is (relaxed) two-special sound, then the resulting NIZK is single-proof extractable [PS00, BN06]. Unfortunately, it is known that in general, this does not hold true in the quantum setting [ARU14]. We define a stronger property for sigma protocol below.

Definition B.3 (Quantum Proof of Knowledge). *A sigma-protocol* Π_Σ *has a* quantum proof of knowledge *with respect to an instance generator* IGen*, if there exists a QPT extractor* $\mathsf{Extract}_\Sigma$*, constants* c_1, c_2*, and polynomial* $p(\lambda)$ *such that for any QPT adversary* $\mathcal{A}$ *(that may output a quantum state* st*) with*

$$\Pr\left[\begin{array}{c} (\mathsf{X}, \mathsf{W}) \xleftarrow{\$} \mathsf{IGen}(1^\lambda) \\ (\alpha, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\mathsf{X}) \\ \beta \xleftarrow{\$} \mathsf{ChSet} \\ \gamma \xleftarrow{\$} \mathcal{A}(\mathsf{X}, \alpha, \beta, \mathsf{st}) \end{array} : \mathsf{Verify}(\mathsf{X}, (\alpha, \beta, \gamma)) = \top\right] \geq \mu(\lambda),$$

we have

$$\Pr\left[\begin{array}{c} (\mathsf{X}, \mathsf{W}) \xleftarrow{\$} \mathsf{IGen}(1^\lambda) \\ \mathsf{W}' \xleftarrow{\$} \mathsf{Extract}_\Sigma^{\mathcal{A}}(\mathsf{X}) \end{array} : (\mathsf{X}, \mathsf{W}') \in \mathcal{R}_{\mathsf{gap}}\right] \geq \frac{1}{p(\lambda)} \cdot \mu(\lambda)^{c_1} - \mathsf{negl}(\lambda),$$

where the runtime of $\mathsf{Extract}_\Sigma$ *is upper bounded by* $c_2 \cdot \mathsf{Time}(\mathcal{A})$ *and we assume one oracle access to* $\mathcal{A}$ *takes* $\mathsf{Time}(\mathcal{A})$.

It was shown in [LZ19, DFMS19] (which was further refined in [DFM20]) that if a sigma protocol is a quantum proof of knowledge, then the Fiat-Shamir transform provides an NIZK that is single-proof extractable even against quantum cheating provers.

Theorem B.4 (Sigma Protocol with QPoK to NIZK with Single-Proof Extractability). *Let us define a slight variant of the single-proof extractability provided in Definition 2.9, where the statement is not quantified for all* $\mathsf{X} \in \mathcal{L}_\mathcal{R}$ *but rather a random* X *sampled by the instance generator* IGen*. Then, if a sigma-protocol* Π_Σ *for relations* $(\mathcal{R}, \mathcal{R}_{\mathsf{gap}})$ *with an associated instance generator* IGen *is a quantum proof of knowledge with parameters* (c_1, c_2) *and* $p(\lambda)$*, then the* NIZK *proof system* Π_{NIZK} *obtained by performing the Fiat-Shamir transform on* Π_Σ *is single-proof extractable in the QROM with parameters* $(c_1, c_2, 2 \cdot c_1)$ *and* $p(\lambda)$.

B.2 Compatible Separable Function

In general, it is not an easy task to check whether a sigma protocol is a quantum proof of knowledge. Liu and Zhandry [LZ19] provided a tool called compatible separable function that allows to prove certain type of sigma protocols to be quantum proofs of knowledge in a "classical" fashion.

Definition B.5 (Compatible Separable Function). *Let Π_Σ be a sigma-protocol for relations $(\mathcal{R}, \mathcal{R}_{\mathsf{gap}})$ with an associated instance generator IGen. Let $(\tau(\lambda), \nu(\lambda))$ be polynomials such that $\tau(\lambda)$ and $\tau(\lambda) - \nu(\lambda)$ are non-negligible. Then, a* (τ, ν)-compatible separable function *for Π_Σ consists of the PPT algorithm $\mathsf{CSF.Gen}$[17] defined as follows:*

$\mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta, \mathsf{mode}) \to f$: *The algorithm, on input the security parameter 1^λ, statement $\mathsf{X} \in \mathcal{L}_\mathcal{R}$, a first flow commitment $\alpha \in \mathsf{ComSet}$, a challenge $\beta \in \mathsf{ChSet}$, and a $\mathsf{mode} \in \{\mathsf{preserving}, \mathsf{separating}\}$, outputs a description of an (classically) efficiently computable function f with binary outputs.*

Moreover, depending on the mode*, we have the following, where $V_{\mathsf{X},\alpha,\beta}$ is defined as the set of all valid third flow $\{\gamma \mid \mathsf{Verify}(\mathsf{X}, (\alpha, \beta, \gamma)) = \top\}$ (which is possibly empty):*

- $(\mathsf{mode} = \mathsf{preserving})$ *For any $\mathsf{X} \in \mathcal{L}_\mathcal{R}$ and $(\alpha, \beta) \in \mathsf{ComSet} \times \mathsf{ChSet}$ such that $|V_{\mathsf{X},\alpha,\beta}| \geq 1$, we have*

$$\Pr\left[f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta, \mathsf{preserving}) : |\{f(\gamma) \mid \gamma \in V_{\mathsf{X},\alpha,\beta}\} = 1|\right] \geq \tau(\lambda).$$

- $(\mathsf{mode} = \mathsf{separating})$ *For any $\mathsf{X} \in \mathcal{L}_\mathcal{R}$, $(\alpha, \beta) \in \mathsf{ComSet} \times \mathsf{ChSet}$, there exists a (possibly negative valued) polynomial $\xi(\lambda)$ such that $\xi(\lambda) \leq \nu(\lambda)$ and for every pair of distinct $\gamma, \gamma' \in V_{\mathsf{X},\alpha,\beta}$, we have*

$$\Pr\left[f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta, \mathsf{separating}) : f(\gamma) = f(\gamma')\right] = \frac{1 + \xi(\lambda)}{2}$$

Definition B.6 (Mode Indistinguishability). *To define mode indistinguishability, we consider the following game between an adversary and a challenger.*

- *The challenger generates $(\mathsf{X}, \mathsf{W}) \xleftarrow{\$} \mathsf{IGen}(1^\lambda)$ and sends X to $\mathcal{A}$.*
- *$\mathcal{A}$ sends a pair $(\alpha, \beta) \in \mathsf{ComSet} \times \mathsf{ChSet}$ to the challenger.*
- *The challenger choses a random bit $\mathsf{coin} \xleftarrow{\$} \{0, 1\}$ and gives $\mathcal{A}$ the function $f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta,$ $\mathsf{preserving})$ if $\mathsf{coin} = 0$, and $f \xleftarrow{\$} \mathsf{CSF.Gen}(1^\lambda, \mathsf{X}, \alpha, \beta, \mathsf{separating})$ otherwise.*
- *$\mathcal{A}$ outputs its guess coin'.*

We say that $\mathcal{A}$ wins if $\mathsf{coin} = \mathsf{coin}'$. We say that a (τ, ν)-compatible separable function is mode indistinguishable *if for any QPT adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{mode}}_{\Pi_\Sigma}(\mathcal{A}) := \left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right| = \mathsf{negl}(\lambda).$$

The following proves that a sigma protocol with a compatible separable function is a quantum proof of knowledge. Combining this with Theorem B.4, it suffices to show that a sigma protocol has a compatible separable function to check if the resulting NIZK is single-proof extractable in the QROM. The following is a compilation of [LZ19, Lemma 1, Lemma 3, Theorem 1, Theorem 2]. Note that our definition of a sigma protocol is akin to the definition of an identification protocol in [LZ19] since we consider an instance generator IGen. However, unlike identification protocols, we require a proof of knowledge (thus the following is a result of merging the proof of [LZ19, Theorem 1, Theorem 2]).

Theorem B.7. *Let Π_Σ be a sigma protocol for relations $(\mathcal{R}, \mathcal{R}_{\mathsf{gap}})$ with an associated instance generator IGen and a (τ, ν)-compatible separable function, where $(\tau(\lambda), \nu(\lambda))$ are functions such that $\tau(\lambda)$ and $\tau(\lambda) - \nu(\lambda)$ are non-negligible. Then, if Π_Σ has relaxed two-special soundness, then it is a quantum proof of knowledge with respect to IGen, where $(c_1, c_2) = (3, 2)$ and $p(\lambda) = \left(\frac{\tau(\lambda) - \nu(\lambda)}{2}\right)^2$.*

[17] The original definition of $\mathsf{CSF.Gen}$ given in [LZ19] also takes as input the witness W. However, we observe that this is not used anywhere in the proof so we intentionally remove it.

C Lattice-based Partially Blind Signature

In this section, we show how to slightly modify our blind signature in Section 3 to turn it into a partially blind signature. The construction is almost identical to our blind signature construction, where the only difference is how we bind the signature to the common message $\gamma \in \mathcal{M}_c$ by a hash function.

C.1 Construction of Partially Blind Signature

Construction. We use all the building blocks provided in Section 3.2 with two minor differences. The hash function H_{M} used to hash messages $\mathsf{M} \in \{0,1\}^*$ to ring elements $h \in R_q$ is modified to take a message and common message pair as input $(\mathsf{M}, \gamma) \in \{0,1\}^* \times \mathcal{M}_c$ instead. Moreover, we introduce a new hash function $\mathsf{H}_{\mathsf{M}c} : \mathcal{M}_c \to R_q$. As with the hash functions in Section 3.2, they are modeled by a random oracle in the security proof with appropriate domain separation.

In the following, we highlight by a red underline the differences between the partially and non-partially blind signature constructions.

$\mathsf{BSGen}(1^\lambda)$: It runs $(\mathbf{a}_1, \mathbf{T}_{\mathbf{a}_1}) \xleftarrow{\$} \mathsf{TrapGen}(1^{k_1 d}, q)$, samples $\mathbf{s} \xleftarrow{\$} [-\Delta_{\mathsf{MLWE}}, \Delta_{\mathsf{MLWE}}]^{(k_1+k_2k_3)}_{\mathsf{coeff}}$ 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 $(\mathsf{com}, \pi^{\mathsf{m}}) \xleftarrow{\$} \rho_1$ 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

$$[\mathbf{a}_1 \mid \mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}] \cdot \mathbf{e}^\top = \underline{u - \mathsf{H}_{\mathsf{M}c}(\gamma)}, \tag{21}$$

using $\mathbf{e} \xleftarrow{\$} \mathsf{SampleLeft}(\mathbf{a}_1, [\mathbf{a}_2 + \mathbf{t} \mid \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_{k_2}], \underline{u - \mathsf{H}_{\mathsf{M}c}(\gamma)}, \mathbf{T}_{\mathbf{a}_1}, \sigma)$. 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 $h = \underline{\mathsf{H}_{\mathsf{M}}(\gamma, \mathsf{M})}$ and $t_i = \mathbf{b}_i \mathbf{r}_i^\top + h \cdot g_i \in R_q$, where t_i and g_i are the i-th entry 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 $\mathbf{e}_3 = [\mathbf{e}_{3,1} \mid \cdots \mid \mathbf{e}_{3,k_2}] \in R^{k_2 k_3}$ and $\mathbf{e}_2 = [e_{2,1} \mid \cdots \mid e_{2,k_2}] \in R^{k_2}$ are parsed intro appropriate sizes. It then creates a proof $\pi^{\mathsf{s}} \xleftarrow{\$} \mathsf{Prove}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, \underline{u - \mathsf{H}_{\mathsf{M}c}(\gamma)}, h), \tilde{\mathbf{e}})$ that proves knowledge of a short vector $\tilde{\mathbf{e}}$. If $\perp \leftarrow \mathsf{Verify}^{\mathsf{H}_{\mathsf{s}}}((\mathbf{a}_1, \mathbf{a}_2, (\mathbf{b}_i)_{i\in[k_2]}, \underline{u - \mathsf{H}_{\mathsf{M}c}(\gamma)}, h), \pi^{\mathsf{s}})$, then it outputs $\Sigma = \perp$. Otherwise, it outputs $\Sigma = \pi^{\mathsf{s}}$ as the signature.

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

Correctness. We omit the proof of the following lemma as it can be argued to be almost identically to Lemma 3.3.

Lemma C.1. *The partial blind signature* Π_{BS} *is correct if* $\sigma > \omega(q^{1/k_1} \cdot \sqrt{\log k_1 d})$, $\forall i \in [3], B^{\mathcal{S}}_{\Sigma,i} = \sqrt{k_i d}\sigma$, $\forall i \in [2], B^{\mathcal{U}}_{\Sigma,i} = B^{\mathcal{S}}_{\Sigma,i}, B^{\mathcal{U}}_{\Sigma,3} = \delta B^{\mathcal{S}}_{\Sigma,2} + B^{\mathcal{S}}_{\Sigma,3}$ *and the two* NIZK*s* $\Pi^{\mathsf{s}}_{\mathsf{NIZK}}$ *and* $\Pi^{\mathsf{m}}_{\mathsf{NIZK}}$ *are correct.*

C.2 Security of Partially Blind Signature

In this section, we show that the partially blind signature satisfies partial blindness under malicious keys and one-more unforgeability. Partial blindness is established by the following theorem.

Theorem C.2. *The blind signature* Π_{BS} *is classically (resp. quantumly) blind under malicious keys if the commitment scheme* Π_{Com} *is classically (resp. quantumly) hiding, and 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 classically (resp. quantumly) zero-knowledge.*

Proof Sketch. Observing that the common message γ is provided in the clear, the proof for partial blindness is almost identical to that of the blind signature (cf. Theorem 3.4 for the classical proof and Theorem 5.1 for the quantum proof). The only difference is that the reduction replaces all occurrence of u by $u - \mathsf{H}_{\mathsf{m}c}(\gamma)$ in the security proof. □

One-more unforgeability is established by the following theorem.

Theorem C.3. *The blind signature* Π_{BS} *is classically (resp. 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 classically (resp. 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. The classical (resp. quantum) proof is almost identical to those in Theorem 3.5 (resp. Theorem 5.2). Below, we provide a proof sketch of the classical proof. We only highlight the games that are different from those in Theorem 3.5, where we further make the assumption that $\mathcal{A}$ makes at most $\mathsf{Q}_{\mathsf{H}_{\mathsf{M}_c}}$ random oracle query to $\mathsf{H}_{\mathsf{M}c}$.

Game_1 **to** Game_3**:** These are defined identically to those of Thm. 3.5 of the full version.

Game_4**:** This is almost identical to that of Thm. 3.5 of the full version. The only difference is that we take into consideration the common message γ. When $\mathcal{A}$ queries (γ', M'_j) as its j-th ($j \in [\mathsf{Q}_{\mathsf{H}_{\mathsf{M}}}]$) random oracle query to H_{M}, the challenger returns h_j. Moreover, at the end of the game, when $\mathcal{A}$ outputs the forgery $\{\gamma, (\mathsf{M}_i, \Sigma_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$, the challenger checks if $(\gamma, \mathsf{M}'_{j^*}) \in \{(\gamma, \mathsf{M}_i)\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$ and if $\{\mathsf{H}_{\mathsf{M}}\gamma, \mathsf{M}_i\}_{i\in[\mathsf{Q}_{\mathsf{S}}+1]}$ are pairwise distinct. Otherwise, the game is identical to that of Thm. 3.5 of the full version. It is easy to check that Lem. 3.7 of the full version holds without any modification.

Game_5 **to** Game_6**:** These are defined identically to those of Thm. 3.5 of the full version.

Game_7**:** This is the only part that deviates from the proof of Thm. 3.5 of the full version. In this game, when $\mathcal{A}$ queries the random oracle $\mathsf{H}_{\mathsf{M}c}$, the challenger samples $\mathbf{s}' \xleftarrow{\$} \chi^{k_1}_{\mathsf{MLWE}}$ and returns $u = \mathbf{a}_1\mathbf{s}'^{\top} \in R_q$ rather than $u \xleftarrow{\$} R_q$. Recalling that $\mathbf{a}_1 = [1 \mid \mathbf{a}'_1] \in R^k_q$ (see Lemma 2.19), it is clear that the Game_6 and Game_7 are indistinguishable assuming the MLWE assumption. Namely, there exists an efficient adversary $\mathcal{B}'_{\mathsf{MLWE}}$ against the MLWE problem such that

$$\Pr[\mathsf{E}_7] \geq \Pr[\mathsf{E}_6] - \mathsf{Q}_{\mathsf{H}_{\mathsf{M}_c}} \cdot \mathsf{Adv}^{\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}}(\mathcal{B}'_{\mathsf{MLWE}})$$

where $\mathsf{Time}(\mathcal{B}'_{\mathsf{MLWE}})$ is roughly $\mathsf{Time}(\mathcal{A}) + \mathsf{Time}(\mathcal{C}_7)$.

Using an almost identical proof to Lemma 3.9, we are able to turn $\mathcal{A}$ in Game_7 into an MSIS solver. The only difference is that we get the following instead of Eq. (9).

$$\forall i \in [3], \|\tilde{\mathbf{e}}_i\|_2 \leq B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i} \ \wedge \ \|c\|_1 \leq B_c \ \wedge \ \left[\mathbf{a}_1 \mid \mathbf{a}_2 + h \cdot \mathbf{g} \mid \widehat{\mathbf{b}}\right] \tilde{\mathbf{e}}^{\top} = c \cdot (u - \mathsf{H}_{\mathsf{M}c}(\gamma)). \tag{22}$$

where γ is the common message included in the forgery. Due to the modification we made in Game_7, $\mathsf{H}_{\mathsf{M}_c}(\gamma) = \mathbf{a}_1\mathbf{s}'^\top$ for some $\mathbf{s}' \in R^{k_1}$ such that $\|\mathbf{s}'\|_2 \leq B_{\mathsf{MLWE}}$. The procedure of extracting an MSIS solution from Eq. (22) is identical to that of Lemma 3.9, where the bound on the extracted solution is increased by $B_c \cdot B_{\mathsf{MLWE}}$ due to $\mathbf{s}'$.

This completes the proof of the classical version of the theorem. We note that the proof of the quantum version is almost identical. The only difference is that we use Lemma 2.26 to program the output of $\mathsf{H}_{\mathsf{M}_c}$ to be MLWE instances rather than uniform random elements over R_q. □

D Reference for Setting the Parameters

We list all the constraints on the various parameters and derive concrete parameters from them in Section 4.5.

Trapdoor-Sampling-Compatible Commitments.

- Correctness: $\delta \geq \sqrt{k_3 d} + \sqrt{k_2 d}$.
- Hiding: $\mathsf{MLWE}_{d,2,k_3-2,S_3,\max(q,q')}$, where S_3 is the uniform distribution over $[-1,1]^{\mathsf{coeff}} \subset R_q$.
- Binding: The requirements on binding are subsumed by the MSIS instance extracted in the multi-proof extractable NIZK.

Single-Proof Extractable NIZK.

- Correctness: $\forall i \in [3]$, $B^{\mathcal{U},\mathsf{gap}}_{\Sigma,i} = 11B_c\sqrt{k_i d}B^{\mathcal{U}}_{\Sigma,i}$.
- Zero-knowledge: Holds statistically.
- Soundness: Subsumed by the constraint for the one-more unforgeability of the blind signature.

Multi-Proof Extractable NIZK.

- Correctness: $B_{\mathbf{Z}} = \sqrt{k_4 d}\gamma_{\mathbf{S}}$, $B_{\mathbf{Z}'} = \sqrt{k_3 d}\gamma_{\mathbf{Y}'}$, $B = \sqrt{d}\gamma_{h'}$, $B_{1,\mathbf{F}} = B_{2,\mathbf{F}} = 12\gamma_{\bar{\mathbf{D}}}$, $B_{1,\mathbf{F}'} = B_{2,\mathbf{F}'} = 12\gamma_{\bar{\mathbf{D}}'}$. With $\gamma_{\mathbf{S}} = 11B_{r,\mathbf{Z}} = 11\sqrt{k_4 k_3 d}\gamma_{\mathbf{E}}$, $\gamma_{\mathbf{Y}'} = 11B_{r,\mathbf{Z}'} = 11B_cB_{\mathbf{R}}$, $\gamma_{h'} = 11B_{r,\zeta} = 11B_cB_h$, $\gamma_{\bar{\mathbf{D}}} = 11B_{r,\mathbf{F}} = 11\sqrt{k_4 k_3 d}\gamma_{\mathbf{D}}$, $\gamma_{\bar{\mathbf{D}}'} = 11B_{r,\mathbf{F}'} = 11B_c\sqrt{k_3 k_3 d}\gamma_{\mathbf{D}'}$. Where $B_{\mathbf{R}}$ and B_h are upper bounds on the norm of $\mathbf{R}$ and h and can be taken as $\sqrt{k_2 k_3 d}$ and $2\sqrt{d}$ to be always true or can be smaller if we assume that the prover samples $\mathbf{R}$ and h until they are below the appropriate bounds. We consider $B_{\mathbf{R}} = \sqrt{k_2 d} + \sqrt{k_3 d}$, $B_h = \sqrt{d}$, we verify experimentally that the prover has probability more than $1/2$ that both of these bounds are correct. We also require δ^{gap} to be an upper bound on the spectral norm of any ternary matrix in $R^{k_3 \times k_2}$, hence $\delta^{\mathsf{gap}} = \sqrt{k_2 k_3 d}$.
- Zero-knowledge: $\mathsf{MLWE}_{d,4k_3+1,k_4-(4k_3+1),\gamma_{\mathbf{E}},Q}$, $\mathsf{MLWE}_{d,1,1,\gamma_{\bar{\mathbf{D}}},Q}$, $\mathsf{MLWE}_{d,1,\kappa,\gamma_{\bar{\mathbf{D}}'},Q}$, $\mathsf{MLWE}_{d,1,\kappa,\gamma_{\bar{d}'},Q}$.
- Soundness: $\mathsf{DSMR}_{d,1,\chi_{\mathsf{DSMR}},Q,p}$. $\mathsf{MSIS}_{d,1,k_4,16B_{\mathbf{Z}},q'}$. $\mathsf{MSIS}_{d,1,k_3,2(B_{\mathbf{Z}'}+B_c\delta^{\mathsf{gap}}),q'}$. The decryption of the NTRU encryption requires that $\|pv\mathbf{F}'_1 + pf\mathbf{F}'_2\|_\infty < \frac{Q}{2}$ where $H = pv/f$ with $f, v \xleftarrow{\$} D_{\gamma_{\mathsf{DSMR}}}$. Hence we need $Q > 4d(B_{1,\mathbf{F}'} + B_{2,\mathbf{F}'})p\gamma_{\mathsf{DSMR}}$ (similarly we need $Q > 4d(B_{1,\mathbf{F}'} + B_{2,\mathbf{F}'})p\gamma_{\mathsf{DSMR}}$). For correct decryption we also require that $\|v\mathbf{Z}'\|_\infty < \frac{p}{2}$, hence we take $p > 5\gamma_{\mathsf{DSMR}}\sqrt{k_3 d}B_{\mathbf{Z}'}$ (similarly we need $p > 5\gamma_{\mathsf{DSMR}}\sqrt{k_4 d}B_{\mathbf{Z}}$).

Blind Signature.

- Trapdoor sampling: $\sigma = \max(1.17\eta_\epsilon q^{\frac{1}{k_1}}, 1.17\eta_\epsilon q^{\frac{1}{k_3}}, \eta_\epsilon\sqrt{\delta^2_{\mathsf{MLWE}} + \delta^2_{\mathsf{gap}}}q^{1/k_2})$. With $\delta_{\mathsf{MLWE}} = \gamma_{\mathsf{MLWE}}(\sqrt{k_1 d} + \sqrt{k_2 d})$, $\delta_{\mathsf{gap}} = \sqrt{k_2 k_3 d}$ and η_ϵ the smoothing parameter which we consider to be $\eta_\epsilon = 2$ using the analysis of e.g., [CPS+20] .
- Relation between $\mathbf{e}$ and $\tilde{\mathbf{e}}$: $\forall i \in [2], B^{\mathcal{U}}_{\Sigma,i} = B^{\mathcal{S}}_{\Sigma,i}$, $B^{\mathcal{U}}_{\Sigma,3} = \delta B^{\mathcal{S}}_{\Sigma,2} + B^{\mathcal{S}}_{\Sigma,3}$. This results in the bounds on the norm of $\mathbf{e}$: $\forall i \in [3]$, $B^{\mathcal{S}}_{\Sigma,i} = \sqrt{k_i d}\sigma$.

- Blindness: Holds based zero-knowledge of the single-proof and multi-proof extractable NIZKs.
- One-more unforgeability: $\mathsf{MSIS}_{d,1,k_1+k_3,B_{\mathsf{MSIS}}}$, with $B_{\mathsf{MSIS}} = B_{\Sigma,1}^{\mathcal{U},\mathsf{gap}} + \delta B_{\Sigma,2}^{\mathcal{U},\mathsf{gap}} + B_{\Sigma,3}^{\mathcal{U},\mathsf{gap}} + B_c\Delta(\sqrt{k_1 d} + \sqrt{k_3 d})$.
- Lemma 3.9: $\mathsf{MLWE}_{d,1,k_1-1,\chi_{\mathsf{MLWE}},q}$, $\mathsf{DSMR}_{d,k_3,\chi_{\mathsf{DSMR}},q}$.

Note that since $\mathbf{b}_1 = [0|1|\mathbf{b}_1'] \in R_q^{k_3}$, for many parameters we can consider the dimension of $\mathbf{b}_1$ to be $k_3 - 1$ instead of k_3 (e.g. when bounding $s_1(\mathbf{R})$). This is reflected in the chosen parameters.