A New Simple Technique to Bootstrap Various Lattice Zero-Knowledge Proofs to QROM Secure NIZKs

Shuichi Katsumata[1]

[1]AIST, Tokyo, Japan

shuichi.katsumata@aist.go.jp

Abstract

Many of the recent advanced lattice-based Σ-/public-coin honest verifier (HVZK) interactive protocols based on the techniques developed by Lyubashevsky (Asiacrypt'09, Eurocrypt'12) can be transformed into a non-interactive zero-knowledge (NIZK) proof in the random oracle model (ROM) using the Fiat-Shamir transform. Unfortunately, although they are known to be secure in the *classical* ROM, existing proof techniques are incapable of proving them secure in the *quantum* ROM (QROM). Alternatively, while we could instead rely on the Unruh transform (Eurocrypt'15), the resulting QROM secure NIZK will incur a large overhead compared to the underlying interactive protocol.

In this paper, we present a new simple semi-generic transform that compiles many existing lattice-based Σ-/public-coin HVZK interactive protocols into QROM secure NIZKs. Our transform builds on a new primitive called *extractable linear homomorphic commitment* protocol. The resulting NIZK has several appealing features: it is not only a proof of knowledge but also straight-line extractable; the proof overhead is smaller compared to the Unruh transform; it enjoys a relatively small reduction loss; and it requires minimal background on quantum computation. To illustrate the generality of our technique, we show how to transform the recent Bootle et al.'s 5-round protocol with an exact sound proof (Crypto'19) into a QROM secure NIZK by increasing the proof size by a factor of 2.6. This compares favorably to the Unruh transform that requires a factor of more than 50.

Contents

1 Introduction **3**
- 1.1 Our Contribution . . . 4
- 1.2 Technical Overview . . . 5
- 1.3 Related Work . . . 9
- 1.4 Open Problems . . . 9

2 Preliminary **10**
- 2.1 Σ-Protocol . . . 10
- 2.2 Quantum Background . . . 12
- 2.3 Lattices . . . 13

3 Extractable Linear Homomorphic Commitment Protocol **15**
- 3.1 Definition . . . 15
- 3.2 Simplified Definition of Extractable LinHC . . . 18
- 3.3 Interlude: Extractable LinHC Specialized for Lattices . . . 19
- 3.4 First Construction of Extractable LinHC: Only MLWE . . . 20
- 3.5 Second Construction of Extractable LinHC: MLWE + DSMR . . . 24
- 3.6 Downgrading to Simplified/Classical Extractable LinHC for Tighter Proofs . . . 26

4 How to Use Extractable LinHC **27**
- 4.1 Lyubashevsky's Σ-Protocol ⇒ Quantum Secure Σ-Protocol via Simplified Extractable LinHC 27
- 4.2 Lyubashevsky's Σ-Protocol ⇒ QROM Secure Signature via Extractable LinHC and Fiat-Shamir 29

5 Application: Quantum Secure 5-Round Public-Coin Exact Sound Proof and QROM Secure Exact Sound NIZK **34**
- 5.1 Quantum Secure Exact Sound Interactive Proof via Simplified Extractable LinHC . . . 34
- 5.2 QROM Secure Exact Sound NIZK via Extractable LinHC and Fiat-Shamir . . . 38
- 5.3 Candidate Parameters and Comparison . . . 39
- 5.4 Further Applications of Extractable LinHC . . . 40

A Omitted Preliminary **47**
- A.1 The MSIS Assumption . . . 47
- A.2 Classical Lyubashevsky's Σ-Protocol for a Basic Lattice Relation . . . 47
- A.3 Background on Signature Scheme . . . 47
- A.4 Background on Commitment Scheme . . . 48

B Omitted Details from Section 5 **50**
- B.1 Recap: Exact Sound Proof by [BLS19] . . . 50
- B.2 Setting the Parameter . . . 50

C Recap on Unruh's Transform **51**
- C.1 High Level Idea of Unruh's Transform . . . 51
- C.2 Two Reasons for Inefficiency . . . 53
- C.3 Applying the Unruh Transform to Bootle et al's Protocol . . . 54

1 Introduction

The Fiat-Shamir transform [FS87] is one of the most popular methods to construct non-interactive zero-knowledge (NIZK) proofs[1] in the random oracle model (ROM) based on a Σ-protocol (or more generally a public-coin honest-verifier zero-knowledge (HVZK) interactive protocol). Due to the ever-growing risk of quantum computers, understanding the *quantum* security of NIZKs in the *quantum* ROM [BDF+11] based on the Fiat-Shamir transform (or related transforms) have been considered to be an important research topic both in theory and practice. However, although many techniques in the QROM have accumulated in the last decade, including but not limited to [BDF+11, Zha12b, Unr12, BZ13, Unr15, Unr17, KLS18, Zha19, DFMS19, LZ19, DFM20], our understanding of NIZKs in the QROM is still not as clear as those in the classical ROM. Notably, many of the recent lattice-based Σ-/public-coin HVZK interactive protocols, such as [BDL+18, BBC+18, BLS19, YAZ+19, ESLL19, ALS20], based on the techniques developed by Lyubashevsky [Lyu09, Lyu12] fall into the following situations:

- they are not known to be (in)secure when applied the Fiat-Shamir transform in the QROM, and/or
- they can be transformed into a QROM secure NIZK using the Unruh transform [Unr15] but incurs a large overhead, say at least ×50, compared to the underlying interactive protocol.

Considering that we can securely apply the Fiat-Shamir transform to these protocols in the classical ROM to obtain efficient NIZKs, the current state-of-the-affair is unsatisfactory. Below, we briefly recall NIZKs in the QROM.

QROM secure NIZKs. Broadly speaking, there are two breeds of transformation to obtain QROM secure NIZKs (that are a proof *of knowledge*) from a Σ-/public-coin HVZK interactive protocol. One is the Fiat-Shamir transform [FS87] and the other is the Unruh transform [Unr15].

Recently, Don et al. [DFMS19] and Liu and Zhandry [LZ19] showed how to argue security of the Fiat-Shamir transform in the QROM in two steps: they first showed that the Fiat-Shamir transform converts a standard Σ-protocol that is additionally a *quantum proof of knowledge* into an NIZK secure in the QROM, and then additionally showed how to construct a Σ-protocol that is a quantum proof of knowledge. Let us call such a Σ-protocol as a *quantum secure* Σ-protocol. It was shown in [LZ19] (and partially in [DFMS19]) that Lyubashevsky's Σ-protocol for proving possession of a short vector $\mathbf{e}$ such that $\mathbf{Ae} = \mathbf{u}$ is quantum secure for appropriate parameters. Concretely, by increasing the parameters compared to those required by the classically secure protocol, they showed that Lyubashevsky's Σ-protocol has a "collapsing" property. However, such techniques for proving that a Σ-protocol is quantum secure are still limited and it seems non-trivial to generalize them to work for the recent more advanced lattice-based protocols. Moreover, these techniques that require rewinding quantum adversaries so far incur a large reduction loss of at least a factor Q^{4t-2}, where Q is the number of adversarial random oracle queries and t is the number of valid transcripts required to invoke special soundness of the underlying Σ-protocol. Since setting the parameters without taking these huge reduction losses into consideration sometimes lead to concrete attacks [KM07, KZ20], having a tighter reduction is desirable. Recently, Don et al. [DFM20] generalized the first step above to $(2n+1)$-round public-coin HVZK interactive protocols.

On the other hand, Unruh [Unr15] showed an elegant transform that converts any standard Σ-protocol into a QROM secure NIZK. The benefit of the Unruh transform is that it works for any Σ-protocol, the reduction loss is tight, and it is also *straight-line extractable*.[2] The last strong property guarantees that the witness from a proof can be extracted without rewinding the adversary and is especially suitable for applications requiring multiple concurrent executions of NIZKs such as group signatures [BMW03] and anonymous attestations [BCC04]. On the other hand, one of the main downsides is that it may incur a noticeable overhead in the proof size compared to the Fiat-Shamir transform since the transformation crucially relies on the challenge set being small. While the overhead can be reasonable when the underlying Σ-protocol already has a small challenge set, e.g., [CDG+17], it becomes prohibitively large as the challenge set grows. (See Appendix C for

[1] We may simply refer to NIZK proofs or NIZK proofs of knowledge as NIZKs when the distinction is not relevant.
[2] This notion is also called *online* extractable in the literature.

a minimal background on the Unruh transform and its overhead). Recently, Chen et al. [CHR+18] extended the Unruh transform to work against a 5-round public-coin HVZK interactive protocol when restricting the second challenge to be *binary*.

Coming back to lattice-based ZK proofs. There are two main approaches in the current literature to construct lattice-based NIZKs. One builds on the Fiat-Shamir with abort paradigm developed by Lyubashevsky [Lyu09, Lyu12] and the other builds on Stern's protocol [Ste94, KTX08]. While the QROM security of the latter approach is well understood since it has a simple combinatorial "commit-and-open" structure [DFMS19, ?], the QROM security of the former approach remains elusive. Notably, for the recent lattice-based protocols such as [BDL+18, BBC+18, BLS19, YAZ+19, ESLL19, ALS20], we either still do not know how to apply the Fiat-Shamir transform and/or require to pay a huge overhead when adopting the Unruh transform to argue QROM security. Therefore, a natural question is:

> *Can we generically and more efficiently transform lattice-based* Σ*-/public-coin* HVZK *interactive protocols based on the Fiat-Shamir with abort paradigm into QROM secure* NIZKs*?*

Ultimately, we would like the transform to achieve the best of the two known transforms: to maintain similar proof size and soundness error of the underlying Σ-protocol like the Fiat-Shamir transform [FS87], while also providing a tight reduction along with a straight-line extractor like the Unruh transform [Unr15].

1.1 Our Contribution

In this work, we provide partial affirmative answers to the above problem. We present a new simple semi-generic transform that compiles many existing lattice-based Σ-/public-coin HVZK interactive protocols such as [BDL+18, BLS19, YAZ+19, ESLL19, ALS20] into a QROM secure NIZK that is also straight-line (simulation) extractable [FKMV12]. The proof overhead is smaller compared to the Unruh transform and enjoys a relatively small reduction loss. In many cases, the reduction loss only scales linearly with t (i.e., number of valid transcripts to invoke special soundness), rather than exponentially (e.g., Q^{4t-2}) required by the Fiat-Shamir transform explained above. This is quite desirable since t can get quite large in recent advanced protocols; for instance [ALS20] requires $t = 32$ in one of their settings, making the reduction loss as large as 2^{638} for a modest $Q = 2^{20}$.

As a concrete example, we show how to transform the recent Bootle et al.'s 5-round protocol with an exact sound proof [BLS19] into a QROM secure NIZK by only increasing the proof size by a factor of 2.6.[3] This is in contrast to using the recent extended Unruh transform [CHR+18][4], which increases the proof size by a larger factor of 51.8. Note that we are not aware of any method to securely apply the Fiat-Shamir transform to Bootle et al.'s protocol in the QROM. Finally, we highlight that not only our transform is very simple but the security proofs are also quite simple and involves a minimal amount of discussion regarding quantum computation.

Our contribution can be divided into the following steps. We only provide a high-level explanation of each step below and refer to Section 1.2 for a more detailed overview.

1. We first propose a new 3-round public-coin interactive protocol called *extractable linear-homomorphic commitment* (LinHC) protocol. (See Section 3)

2. We then show how to bootstrap a broad class of Σ-protocols into a Σ-protocol that is also a *quantum straight-line proof of knowledge* by using an extractable LinHC protocol. Here, we consider the class of Σ-protocols where the response (i.e., the prover's third message) is of the form $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$, where $\mathbf{e} \in \mathbb{Z}_q^m$ is the witness, β is the challenge sampled by the verifier, and $\mathbf{r} \in \mathbb{Z}_q^m$ is the masking term committed in the prover's first message.[5] (See Section 4.1)

[3] As a point of reference, the signature scheme Dilithium, a finalist to the NIST post-quantum standardization process based on the simple Lyubashevsky's Σ-protocol, requires to increase the sum of public key and signature size by a factor 3.2 to achieve QROM security [KLS18].

[4] Since Bootle et al's protocol requires slightly more transcripts for special soundness compared to those considered in [CHR+18], the security proof of [CHR+18] may need to be modified to apply the transform to Bootle et al's protocol.

[5] Although we consider a slightly broader type of Σ-protocols in the main body, we keep the presentation simple here as the main idea generalizes easily.

3. We further show that we can apply the Fiat-Shamir transform to Σ-protocols with a quantum straight-line proof of knowledge to construct a QROM secure NIZK that is also straight-line extractable. (See Section 4.2)

4. We provide two simple constructions of lattice-based extractable LinHC protocols: one based on the module learning with errors (MLWE) problem, and the other based on the MLWE *and* the decisional small matrix ratio (DSMR) problem, where the latter is more efficient. Here the DSMR problem is a generalization of the decisional small polynomial ratio problem [LTV12, SXY18] defined over a module NTRU lattice [**?**]. (See Sections 3.4 and 3.5)

5. Finally, we discuss how to apply extractable LinHC protocols to more advanced lattice-based public-coin HVZK interactive protocols. As a concrete example, we provide the details on how to make Bootle et al.'s 5-round protocol with an exact sound proof [BLS19] into a QROM secure NIZK with concrete parameters. We chose this protocol since it is one of the more complex protocols that have appeared in the literature while still being relatively simple enough to fit in our framework. We show how the ideas can be used to obtain similar results for other protocols such as [BDL+18, YAZ+19, ESLL19, ALS20]. (See Section 5)

One notable difference between our transform and prior transforms that achieve straight-line extractable NIZKs either in the classical or post-quantum setting (i.e., Fischlin [Fis05] and Unruh [Unr15]) is that ours do not put any restriction on the size of the challenge set of the underlying Σ-protocol. Therefore, if the underlying Σ-protocol has an exponentially large challenge set, we can use it directly to obtain an NIZK, thus circumventing an inefficient soundness amplification required by prior transforms. We note that our result does not contradict the impossibility result of Fischlin [Fis05] who (roughly) showed that an NIZK in the ROM with a straight-line extractor that cannot program the random oracle requires a prover to query the random oracle on at least $\omega(\log \kappa)$ points to produce a proof, where κ is the security parameter.[6] The main reason is that our NIZK requires the extractor to program the (Q)RO similar to the proof in the Fiat-Shamir transform. The difference between the Fiat-Shamir transform is that our extractor reprograms the (Q)RO in a way that it does not require to rewind the adversary to extract the witness.

1.2 Technical Overview

We provide an overview of each step explained in the above contribution.

Items 1 and 2: Extractable LinHC protocols and integrating it to Σ-protocols. We use Lyubashevsky's Σ-protocol [Lyu09, Lyu12], which we denote by Σ_{Lyu}-protocol, as a leading example. It forms the basis of lattice-based zero-knowledge proofs based on the Fiat-Shamir with abort paradigm and the ideas presented below extend naturally to more advanced protocols.

Let $\mathbf{A} \in R_q^{n \times m}$ and $\mathbf{u} \in R_q^n$ be public, where R and R_q denote the rings $\mathbb{Z}[X]/(X^d+1)$ and $\mathbb{Z}_q[X]/(X^d+1)$. Then, the Σ_{Lyu}-protocol allows one to prove knowledge of a short vector $\mathbf{e} \in R^m$ satisfying $\mathbf{Ae} = \mathbf{u}$.[7] The prover first sends $\mathbf{w} = \mathbf{Ar}$ to the verifier where $\mathbf{r} \in R^m$ is a random short vector sampled from some specific distribution. The verifier returns a randomly sampled challenge $\beta \leftarrow \{0,1\}^d$, where β is viewed as an element over R by the standard coefficient embedding. Finally, the prover sends $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$ to the verifier. Here, it is standard to perform a rejection sampling step to make $\mathbf{z}$ statistically independent from $\mathbf{e}$. However, we ignore this subtle issue in the overview. Finally, the verifier accepts if $\mathbf{z}$ is short and $\mathbf{Az} = \beta \cdot \mathbf{u} + \mathbf{w}$ holds. It is known that the Σ_{Lyu}-protocol satisfies *relaxed* (rather than *exact*) special soundness: Given two valid transcripts of the form $(\mathbf{w}, \beta, \mathbf{z})$ and $(\mathbf{w}, \beta', \mathbf{z}')$ with $\beta \neq \beta'$, an extractor $\mathsf{Extract}_{\mathsf{ss}}$ outputs a witness $\mathbf{z}^* = \mathbf{z} - \mathbf{z}'$ such that $\mathbf{Az}^* = (\beta - \beta') \cdot \mathbf{u}$. Here, although $\mathbf{z}^*$ does not lie in the original relation, such proof of knowledge for a *relaxed* relation has proven to suffice in many applications.

Modifying the Σ_{Lyu}-protocol. Our idea to turn the Σ_{Lyu}-protocol to be a straight-line proof of knowledge is simple. Here, recall that to show a Σ-protocol is straight-line proof of knowledge, informally we need to

[6] This result informally shows that we need at least $\omega(\log \kappa)$-parallel repetition, assuming that a constant number of hash query is required in each repetition.

[7] All operations with elements over R_q are understood to be performed over mod q.

construct an extractor $\mathsf{SL\text{-}Extract}$ that on input a single valid transcript (and some private information), outputs a witness $\mathbf{z}^*$. As a first step, we let the prover commit to its witness $\mathbf{e}$ and randomness $\mathbf{r}$ by a *linear homomorphic* commitment scheme. The prover outputs $\mathbf{w} = \mathbf{Ar}$ as in the original protocol along with two commitments $\mathsf{com}_{\mathbf{e}} = \mathsf{Com}_{\mathsf{pk}}(\mathbf{e})[\delta_{\mathbf{e}}]$ and $\mathsf{com}_{\mathbf{r}} = \mathsf{Com}_{\mathsf{pk}}(\mathbf{r})[\delta_{\mathbf{r}}]$, where pk is a commitment key, and $\delta_{\mathbf{e}}$ and $\delta_{\mathbf{r}}$ are commitment randomness.[8] Then, given a random challenge β from the verifier, the prover returns $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$ *and* the commitment randomness $\delta_{\mathbf{z}} := \beta \cdot \delta_{\mathbf{e}} + \delta_{\mathbf{r}}$ as the third message. The verifier accepts if $\mathbf{z}$ is short; $\mathbf{Az} = \beta \cdot \mathbf{u} + \mathbf{w}$ holds; *and* $\mathsf{Com}_{\mathsf{pk}}(\mathbf{z})[\delta_{\mathbf{z}}] = \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$ holds. Here, for correctness to hold, we require the commitment scheme to satisfy linear homomorphism also over the randomness, i.e., $\beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}} = \mathsf{Com}_{\mathsf{pk}}(\beta \cdot \mathbf{e} + \mathbf{r})[\beta \cdot \delta_{\mathbf{e}} + \delta_{\mathbf{r}}]$ for any $\beta \in \{0,1\}^d \subset R$.

We first check our modified Σ_{Lyu}-protocol remains secure in the standard sense. Special soundness follows since two valid transcripts of the modified Σ_{Lyu}-protocol include two valid transcripts of the original Σ_{Lyu}-protocol. Next, assume $\delta_{\mathbf{z}}$ does not leak any information on the original commitment randomness $\delta_{\mathbf{e}}$ and $\delta_{\mathbf{r}}$. Then, (roughly) we can invoke the hiding property of the commitment scheme to argue that $\delta_{\mathbf{z}}$, $\mathsf{com}_{\mathbf{e}}$, and $\mathsf{com}_{\mathbf{r}}$ leak no information on $\mathbf{e}$ and $\mathbf{r}$ expect that they satisfy $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$. Therefore, since the Σ_{Lyu}-protocol is HVZK, so is our modified Σ_{Lyu}-protocol.

How to extract a witness. To show that it is a straight-line proof of knowledge, we enhance the linearly homomorphic commitment scheme to be *extractable*. Namely, we assume there exists an alternative key generation algorithm $\mathsf{SimKeyGen}$ that outputs a simulated commitment key pk^* with an associated trapdoor τ with the following properties: pk^* is indistinguishable from pk output by the honest key generation algorithm KeyGen, and there exists a commitment extractor $\mathsf{Extract}_{\mathsf{Com}}$ such that on input the trapdoor τ and an honestly generated commitment $\mathsf{com}_{\mathbf{x}} = \mathsf{Com}_{\mathsf{pk}^*}(\mathbf{x})[\delta_{\mathbf{x}}]$, outputs $\mathbf{x}$. Intuitively, it seems such an extractor $\mathsf{Extract}_{\mathsf{Com}}$ immediately implies a straight-line extractor $\mathsf{SL\text{-}Extract}$. On input a valid transcript $((\mathbf{w}, \mathsf{com}_{\mathbf{e}}, \mathsf{com}_{\mathbf{r}}), \beta, (\mathbf{z}, \delta_{\mathbf{z}}))$, $\mathsf{SL\text{-}Extract}$ just runs $\mathbf{e} \leftarrow \mathsf{Extract}_{\mathsf{Com}}(\tau, \mathsf{com}_{\mathbf{e}})$ to extract the witness $\mathbf{e}$. However, this intuition is clearly wrong since an adversary might have constructed a *malformed* commitment $\mathsf{com}_{\mathbf{e}}$ and $\mathsf{com}_{\mathbf{r}}$ that satisfies $\mathsf{Com}_{\mathsf{pk}^*}(\mathbf{z})[\delta_{\mathbf{z}}] = \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$. Notably, the only commitment $\mathsf{SL\text{-}Extract}$ sees that is guaranteed to be valid is $\beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$ due to correctness. However, since $\mathsf{SL\text{-}Extract}$ already knows that this opens to $\mathbf{z}$, there seems to be no point using the trapdoor τ.

The main observation here is that since the adversary must prepare $\mathsf{com}_{\mathbf{e}}$ and $\mathsf{com}_{\mathbf{r}}$ *before* seeing the challenge β, there should be several other β's in $\{0,1\}^d$ that it would have been able to produce valid openings to. To make the discussion simple, we first assume the case where the challenge space of the Σ_{Lyu}-protocol is only of polynomial size and the existence of another valid commitment $\beta' \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$ with $\beta' \neq \beta$ is guaranteed. Then, $\mathsf{SL\text{-}Extract}$ runs through all $\beta \in \{0,1\}^d$ and executes $\mathsf{Extract}_{\mathsf{Com}}(\tau, \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}})$ in polynomial time. Since $\beta' \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$ is guaranteed to be a valid commitment, $\mathsf{Extract}_{\mathsf{Com}}$ outputs the corresponding message $\mathbf{z}'$ committed to $\beta' \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$. After finding such $\mathbf{z}'$, $\mathsf{SL\text{-}Extract}$ can invoke the special soundness extractor $\mathsf{Extract}_{\mathsf{ss}}$ on input $(\mathbf{w}, \beta, \beta', \mathbf{z}, \mathbf{z}')$ to obtain a witness $\mathbf{z}^*$ for the (relaxed) relation. We can turn this rough idea into a formal proof by performing parallel repetition of the Σ_{Lyu}-protocol to amplify the soundness error to be negligible while noticing that $\mathsf{SL\text{-}Extract}$ still only needs to invoke $\mathsf{Extract}_{\mathsf{Com}}$ a polynomial time. However, recall the goal was to extract without having to restrict the challenge space of the Σ_{Lyu}-protocol to be polynomial size as required by the Fischlin and Unruh transforms [Fis05, Unr15].[9]

Making the challenge set exponentially large. By slightly refining the above argument, we can make sure the above idea works even when the challenge set is exponentially large. Assume an adversary has a non-negligible probability ϵ in completing the Σ_{Lyu}-protocol with an honest verifier. Then conditioning on the adversary succeeding, a standard statistical argument shows that with probability at least $1/2$, the adversary must have been able to output a valid response for at least ϵ-fraction of the challenges. That is, there exists $2^d \cdot \epsilon$ other β's in $\{0,1\}^d$ that the adversary was able to output a valid third message $(\mathbf{z}, \delta_{\mathbf{z}})$. Therefore, we define the $\mathsf{SL\text{-}Extract}$ to execute $\mathsf{Extract}_{\mathsf{Com}}(\tau, \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}})$ on roughly (κ/ϵ)-randomly chosen β's. Then, with probability at least $1 - 2^{-\kappa}$, $\mathsf{SL\text{-}Extract}$ finds the desired $\mathbf{z}'$ and the rest follows the same argument made above.

[8] For any probabilistic algorithm $\mathcal{A}$, $\mathcal{A}(x)[\rho]$ denotes running $\mathcal{A}$ on input x with randomness ρ.

[9] To be precise, [Fis05] can use any Σ-protocol with an exponential challenge set size. Nevertheless, it still needs to rely on parallel repetition to amplify soundness since it can only use polynomially of the challenges in a meaningful way.

Since the above argument is purely statistical and agnostic to whether the adversary is classical or quantum, the resulting modified Σ_{Lyu}-protocol is by default a *quantum* straight-line proof of knowledge. In Section 3, we formalize the properties required by the underling commitment scheme and define it as a new interactive protocol called the *extractable linear homomorphic commitment* (LinHC) protocol. We note that the extractable LinHC protocol can be naturally plugged into multi-round public-coin HVZK interactive protocols with similar structures. Finally, an acute reader may have noticed that our resulting Σ-protocol is in the common reference string (CRS) model since it requires a commitment key pk. Although this is true in general, for our specific extractable LinHC protocol, the pk can be the output of the (Q)RO on any input of the prover's choice so the resulting Σ-protocol will *not* require any CRS.

Item 3: Applying the Fiat-Shamir transform in the QROM. A quantum straight-line extractable Σ-protocol is particularly quantum secure so we can appeal to recent techniques [DFMS19, LZ19] to transform it into a QROM secure NIZK or a QROM secure signature. However, we can take advantage of the straight-line extractability of the Σ-protocol to provide simpler and tighter proofs. Recall one of the main reasons that made the proof of Fiat-Shamir transform in the QROM difficult when basing on standard Σ-protocols was that there was no easy way to extract a witness from a forged proof output by the adversary. Therefore, by using the straight-line extractor SL-Extract to extract from the forged proof, it seems we can overcome one of the most difficult obstacles. We outline the proof and explain some of the pitfalls. As commonly done in the literature, below we consider the proof for the deterministic signature scheme based on the Fiat-Shamir transform (which captures the essence of a simulation sound/extractable NIZK).[10]

Proof overview. The proof consists of two parts: first show that if the signature scheme is unforgeable against no-message attack (UF-NMA) secure, then it is secure in the standard sense, i.e., unforgeable against chosen message attack (UF-CMA) secure; next, show that if the relation used by the Σ-protocol is hard, then the signature scheme is UF-NMA secure. Here, recall UF-NMA considers the setting where an adversary is not allowed to make any signing queries.

Part 1: UF-NMA to UF-CMA. The first part of the proof follows closely to those given by Kiltz et al. [KLS18] (which themselves follow [Unr15, Unr17]) who showed quantum security of a Fiat-Shamir transformed signature scheme basing on a special type of Σ-protocol (or more specifically a lossy identification protocol). The main observation is that by using the HVZK simulator of the Σ-protocol, we can make the proof *history-free* [BDF+11]. In particular, for each message M, we *deterministically* generate a transcript $(\mathbf{w}_{\mathsf{M}}, \beta_{\mathsf{M}}, \mathbf{z}_{\mathsf{M}})$ of the Σ-protocol using the HVZK simulator run on message-dependent randomness. Since the simulated transcript is determined uniquely by the message, we can program the random oracle H at the beginning of the game *before* invoking the adversary so that $\mathsf{H}(\mathbf{w}\|\mathsf{M})$ outputs β_{M} if and only if $\mathbf{w} = \mathbf{w}_{\mathsf{M}}$. Then, to answer a signature query, the simulator can output the already programmed simulated proof as the signature.

This high-level approach works for Kiltz et al. [KLS18] without complications, however, we encountered a slight issue in our setting. The main difference is that while the Σ-protocol of Kiltz et al. satisfied statistical HVZK, ours is only computational HVZK. Concretely, for our specific instantiation of the extractable LinHC protocol based on the MLWE assumption, we informally need to argue that a superposition of the MLWE samples of the form $\sum_{\mathbf{s}_{\mathsf{M}},\mathbf{s}'_{\mathsf{M}}} |\mathbf{B}\rangle |\mathbf{B}\cdot\mathbf{s}_{\mathsf{M}} + \mathbf{s}'_{\mathsf{M}}\rangle$, where $\mathbf{s}_{\mathsf{M}}, \mathbf{s}'_{\mathsf{M}}$ are random MLWE secrets, is indistinguishable from $\sum_{\mathbf{s}_{\mathsf{M}},\mathbf{s}'_{\mathsf{M}}} |\mathbf{B}\rangle |\mathbf{b}_{\mathbf{s}_{\mathsf{M}},\mathbf{s}'_{\mathsf{M}}}\rangle$, where $\mathbf{b}_{\mathbf{s}_{\mathsf{M}},\mathbf{s}'_{\mathsf{M}}}$ is a random vector. Unfortunately, we were not able to reduce the standard MLWE assumption to such an assumption. Here, roughly, $\mathbf{B}$ corresponds to the commitment key of the extractable LinHC protocol and each $\mathbf{B}\cdot\mathbf{s}_{\mathsf{M}} + \mathbf{s}'_{\mathsf{M}}$ corresponds to the commitment.

To resolve this issue, we tweak the extractable LinHC protocol to use fresh commitment keys $\mathbf{B}_{\mathsf{M}}$ for each message M and provide a slightly more general definition than what we laid out above. In particular, the extractable LinHC protocol we require to construct a QROM secure NIZK/signature needs to have a more general structure compared to those required to construct a Σ-protocol with a quantum proof of knowledge. In Section 3, the latter is referred to as the "simplified" definition. Here, if we only care about the classical setting, then this issue does not appear so we can always rely on the simplified definition for both cases.

Part 2: Straight-line extractable Σ-protocol to UF-NMA. The remaining piece is to show that we can extract a witness from the forgery output by the adversary. The reduction is the same as before: provided a forgery,

[10] Note that considering deterministic signature schemes is w.l.o.g since we can always derandomize the signing algorithm using pseudorandom functions.

the extractor probes many challenges β randomly until $\mathsf{Extract}_{\mathsf{Com}}(\tau, \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}})$ outputs a valid $\mathbf{z}$, where $\mathsf{com}_{\mathbf{e}}$ and $\mathsf{com}_{\mathbf{r}}$ are the commitments of the extractable LinHC protocol included in the adversary's forgery. The main difference is in the analysis of the success probability of such a procedure. Since β is generated as $\mathsf{H}(\cdots \| \mathsf{com}_{\mathbf{e}} \| \mathsf{com}_{\mathbf{r}})$ when applying the Fiat-Shamir transform, the adversary has some control over the β it uses. To make matters worse, it can make quantum queries to H to obtain a superposition of challenges $\sum_{\beta} \alpha_{\beta} |\beta\rangle$. Therefore, we can no longer rely on the simple statistical argument that relied on β being uniformly random. We will show how to upper bound the number of random sampling the extractor must perform before finding a "good" challenge β by using bounds on the generic quantum search problem [Zha12a, HRS16, KLS18].

Item 4: Constructing extractable LinHC protocols. It remains to show how to construct an extractable LinHC protocol based on lattices. The construction is a simple variant of the (dual) Regev public-key encryption scheme [Reg05, GPV08] that is known to be linearly homomorphic. The commitment key is two random matrices $\mathsf{pk} = (\mathbf{A}, \mathbf{B}) \in R_q^{m \times n} \times R_q^{m \times n}$ and commitments to the short vectors $(\mathbf{e}, \mathbf{r}) \in R_q^m \times R_q^m$ are defined as follows for $X \in \{\mathbf{e}, \mathbf{r}\}$:

$$\mathsf{com}_X := \big(p \cdot (\mathbf{A}\mathbf{s}_{X,1} + \mathbf{s}_{X,2}),\ p \cdot (\mathbf{B}\mathbf{s}_{X,1} + \mathbf{s}_{X,3}) + X\big),$$

where p is some odd integer coprime to q and the $\mathbf{s}$'s are commitment randomness sampled from an appropriate domain. Then, for any challenge $\beta \in \{0,1\}^d \subset R$, we can construct a commitment to $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$ by computing $\mathsf{com}_{\mathbf{z}} = \beta \cdot \mathsf{com}_{\mathbf{e}} + \mathsf{com}_{\mathbf{r}}$, which is again of the form $\mathsf{com}_{\mathbf{z}} = \big(p \cdot (\mathbf{A}\mathbf{s}_{\mathbf{z},1} + \mathbf{s}_{\mathbf{z},2}),\ p \cdot (\mathbf{B}\mathbf{s}_{\mathbf{z},1} + \mathbf{s}_{\mathbf{z},3}) + \mathbf{z}\big)$, where $\mathbf{s}_{\mathbf{z},i} = \beta \cdot \mathbf{s}_{\mathbf{e},i} + \mathbf{s}_{\mathbf{r},i}$ for $i \in [3]$. However, we cannot simply output the tuple $(\mathbf{s}_{\mathbf{z},i})_{i \in [3]}$ as the opening of $\mathsf{com}_{\mathbf{z}}$ to the message $\mathbf{z}$ since $\mathbf{s}_{\mathbf{z},i}$ may leak information of $\mathbf{s}_{\mathbf{e},i}$ and $\mathbf{s}_{\mathbf{r},i}$. Instead, we use the rejection sampling technique [Lyu09, Lyu12] and sample each $\mathbf{s}_{\mathbf{r},i}$ for $i \in [3]$ from a slightly wider distribution compared to those of the $\mathbf{s}_{\mathbf{e},i}$'s and only output the tuple $(\mathbf{s}_{\mathbf{z},i})_{i \in [3]}$ with some fixed probability.[11] Effectively, the opening $(\mathbf{s}_{\mathbf{z},i})_{i \in [3]}$ are independent of the $\mathbf{s}_{\mathbf{e},i}$'s. At this point, we can argue $\mathsf{com}_{\mathbf{e}}$ is indistinguishable from random by invoking the MLWE assumption. Moreover, since $\mathsf{com}_{\mathbf{r}} = \mathsf{com}_{\mathbf{z}} - \beta \cdot \mathsf{com}_{\mathbf{e}}$, we conclude that we can simulate $\mathsf{com}_{\mathbf{r}}$, $\mathsf{com}_{\mathbf{e}}$, and $(\mathbf{s}_{\mathbf{z},i})_{i \in [3]}$ only using $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$. Finally, extractability follows by switching the commitment key pk to be the real public-key of the encryption scheme. We set $\mathsf{pk}^* = (\mathbf{A}, \mathbf{B})$, where $\mathbf{B} = \mathbf{D}_1\mathbf{A} + \mathbf{D}_2$ for two matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ with small entries. Then, for an appropriate set of parameters, given $\mathsf{com}_{\mathbf{z}} = (\mathbf{t}_1, \mathbf{t}_2)$, we can decrypt it by $(\mathbf{t}_2 - \mathbf{D}_1\mathbf{t}_1) \mod p = \mathbf{z}$.

Item 5: A concrete example. Finally, we provide a more interesting use-case for our extractable LinHC protocol other than the Lyubashevsky's Σ-protocol explained above. We consider the 5-round public-coin HVZK interactive protocol by Bootle et al. [BLS19] that achieves *exact* special soundness. So far, we do not know how to apply the Fiat-Shamir transform securely in the QROM to this protocol since unlike the Lyubashevsky's Σ-protocol, there is no natural notion of "collapsingness" [LZ19, DFMS19]. We can instead try applying the recent Unruh transform extended to 5-round protocols by Chen et al. [CHR+18] by limiting the second challenge used by the verifier to be binary. For completeness, we show in Appendix C.3 that assuming the extended Unruh transform applies to Bootle et al's protocol, we incur a factor 51.8 blowup in the proof size. In Section 5, we show that our extractable LinHC works simply as a wrapper and bootstraps the original protocol of Bootle et al. to be quantum secure with an overhead of only a factor 2.6. We also discuss how the same ideas are applicable to other lattice-based protocols such as [BDL+18, YAZ+19, ESLL19, ALS20]. As the main focus of this study is to introduce new theoretical tools and ideas to transform Σ-protocols into QROM secure NIZKs, we leave optimization and assessment of the concrete security of other lattice-based protocols as future work. Finally, we note that applying our extractable LinHC on Lyubashevsky's Σ-protocol does not result in a more efficient QROM secure signature scheme compared to the QROM secure $\mathsf{Dilithium}$ proposed in [KLS18]. Roughly, this is because when viewed as an NIZK, ours achieve a stronger property: while [KLS18] only achieves soundness, we also achieve (straight-line) proof of knowledge.

[11] We ignore in the overview the fact that our extractable LinHC protocol has non-negligible correctness error as it is standard in lattice-based Σ-protocols.

1.3 Related Work

Σ-protocols and NIZKs. Lindell [Lin15] and Ciampi et al. [CPSV16] consider using a dual mode commitment to commit to the first prover message in any Σ-protocol and then to apply the Fiat-Shamir transform. They show the resulting NIZK is sound (and not a proof of knowledge) and satisfies zero-knowledge in the *non-programmable* classical ROM. Similar to ours, Maurer [Mau15] abstracts and formalizes many existing natural Σ-protocols (e.g., Schnorr, Guillou-Quisquater) that admit a linearly homomorphic property in the prover's response. Finally, linearly homomorphic encryption has been used along with the class of Σ-protocols considered in this work to construct a *designated-verifier* NIZK in the standard model, e.g., [DFN06, CC18].

Other lattice-based ZK proofs. Stern's protocol is another starting point to construct lattice-based ZK proofs. It has been extensively used to construct primitives such as ring signatures, group signatures, and e-cash systems [LNSW13, LLM+16, LLNW16, LLNW17]. Recently, Beullens [Beu20] generalized the idea by Katz et al. [KKW18] and constructed more efficient NIZKs based on Stern-type protocols and showed that it is QROM secure if the hash function it uses is collapsing [DFMS19, DFM20]. Bootle et al. [BLNS20] presented the first poly-logarithmic lattice-based ZK proof improving upon [BBC+18]. Lyubashevsky and Neven [LN17] construct a verifiable encryption scheme by proving validity of the ciphertext using NIZKs based on the Fiat-Shamir with abort technique. The decryption algorithm works by (informally) searching through a set of possible valid ciphertext and runs in expected polynomial time. Although this is similar to our straight-line extractor in the sense that they both search for a valid ciphertext/commitment, we believe the resemblance is only superficial. Results by [BLNS20, LN17] are provided in the classical ROM.

QROM secure signatures using the Fiat-Shamir or Unruh transform. Picnic [CDG+17] is based on an identification protocol constructed using the "MPC-in-the-head" technique [IKOS07] and the Unruh transform. Dilithium is based on Lyubashevsky's identification protocol and was shown by Kiltz et al. [KLS18] to enjoy QROM security via the Fiat-Shamir transform by using larger parameters than compared to those required in the classical setting. The result relies on the identification protocol having a lossy key generation. El Kaafarani et al. [EKP20] constructs a signature scheme based on the CSIDH assumption building on the same technique. MQDSS [CHR+16] is based on a multivariate quadratic 5-round identification protocol and Chen et al. [CHR+18] showed QROM security by extending the Unruh transform. Beullens [Beu20] provides a more efficient multivariate quadratic-based signature using the Fiat-Shamir transform due to [DFMS19, **?**]. Recently, Kales and Zaverucha [KZ20] proposed attacks on some of the signature schemes based on a 5-round identification protocol. However, the attack does not contradict the security proof of the original schemes as the schemes were not set based on provably secure parameters.

Concurrent and independent work. Esgin et al. [ENS20] provide a zero-knowledge protocol for exact proofs based on lattices using ideas from [ALS20]. Compared to Bootle et al's protocol [BLS19] they achieve a proof size smaller by a factor 8. Vadim et al. [LNS20] provide an efficient zero-knowledge protocol for proving additive and multiplicative relations of committed integers. Both works use optimizations specific to polynomial rings and can be turned into classical ROM secure NIZKs via the Fiat-Shamir transform. Very recently, in an exciting work by [**?**] showed a general method to rewind quantum adversaries against a collapsing public-coin interactive protocol without incurring an exponential loss in the number of performed rewinding. We believe this technique can be used to make the reduction loss of lattice-based protocols requiring many rewinding, conditioned on that they can be further shown to be collapsing.

1.4 Open Problems

In this work, we provide a new path to modify many of the lattice-based Σ-protocols into QROM secure NIZKs using the Fiat-Shamir transform. Many of the recent efficient lattice-based Σ-protocols uses optimizations exploit the algebraic structure of the polynomial rings. We leave it as an interesting open problem whether we can optimize our extractable LinHC using similar techniques to achieve a transform that incurs less overhead than what we propose. Another important problem is to figure out whether there is an efficient way to convert these lattice-based Σ-protocol to be "collapsing" so that we can securely apply the Fiat-Shamir transform [DFMS19, LZ19, DFM20] in the QROM.

2 Preliminary

Notation. For a set S and distribution (or algorithm) D, "$\leftarrow S[\rho]$" and"$\leftarrow D[\rho]$" denote the process of uniformly sampling from S with randomness ρ and sampling from (or executing) D with randomness ρ, respectively. For sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathsf{Func}(\mathcal{X},\mathcal{Y})$ denotes the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. For column vectors $\mathbf{a}$ and $\mathbf{b}$, $[\mathbf{a}\|\mathbf{b}]$ denotes the vertical concatenation. With an overload of notation, for two strings s and r over some alphabet, $s\|r$ denotes the concatenated string. We use PPT and QPT as shorthand for probabilistic polynomial time and quantum polynomial time, respectively.

2.1 Σ-Protocol

Let $\mathcal{R} \subset \{0,1\}^* \times \{0,1\}^*$ be a polynomial time recognizable relation. For $(\mathsf{X},\mathsf{W}) \in \mathcal{R}$, we call X as the statement and W as the witness. Furthermore, let $\mathcal{L} = \{\mathsf{X} \mid \exists \text{ s.t. } (\mathsf{X},\mathsf{W}) \in \mathcal{R}\}$ be the corresponding **NP** language to the relation $\mathcal{R}$.

A Σ-*protocol* defined for a relation $\mathcal{R}$ is a public-coin three-move interactive protocol between a prover and a verifier. As with many lattice-based Σ-protocols, we define a slightly relaxed variant of the standard Σ-protocol where soundness holds for a wider relation $\mathcal{R}'$ than the relation $\mathcal{R}$ being used in the actual protocol. Below, we provide a definition of a Σ-protocol in the *common reference string* model [Bel20].[12] An overview is depicted in Figure 1. We note that this does not loose generality since most of the results known to hold for Σ-protocols in the plain model hold for Σ-protocols in the CRS model. Moreover, in many cases, Σ-protocols are implicitly defined in the CRS model by, for example, assuming public parameters for a commitment scheme is provided to the prover and verifier.

Prove: $\mathsf{crs}, (\mathsf{X},\mathsf{W}) \in \mathcal{R}$		Verifier: $\mathsf{crs}, \mathsf{X} \in \mathcal{L}$
$(\alpha, \mathsf{st}) \leftarrow \mathsf{Prove}_1(\mathsf{X},\mathsf{W})$	$\xrightarrow{\alpha}$	
	$\xleftarrow{\beta}$	$\beta \leftarrow \mathsf{ChSet}$
$\gamma \leftarrow \mathsf{Prove}_2(\mathsf{X},\mathsf{W},(\alpha,\beta,\mathsf{st}))$	$\xrightarrow{\gamma}$	
		$\top$ or $\bot \leftarrow \mathsf{Verify}(\mathsf{X},(\alpha,\beta,\gamma))$

Figure 1: Σ-protocol in the CRS model. All the algorithms are assumed to be given crs as input.

Definition 2.1 (Σ-protocol in the CRS model). *A Σ-protocol in the common reference string (CRS) model for relations $(\mathcal{R},\mathcal{R}')$ such that $\mathcal{R} \subseteq \mathcal{R}'$ is defined by a tuple of algorithms* $(\mathsf{Setup},\mathsf{Prove} = (\mathsf{Prove}_1,\mathsf{Prove}_2),\mathsf{Verify})$*, 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 Σ-protocol in the CRS model proceeds as follows:*

1. *A common reference string is sampled* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$*;*
2. *The prover, on input* crs *and* $(\mathsf{X},\mathsf{W}) \in \mathcal{R}$*, runs* $(\alpha,\mathsf{st}) \leftarrow \mathsf{Prove}_1(\mathsf{crs},\mathsf{X},\mathsf{W})$ *and returns* $\alpha \in \mathsf{ComSet}$ *to the verifier;*
3. *The verifier then samples a challenge* $\beta \leftarrow \mathsf{ChSet}$ *and returns it to the prover;*
4. *The prover sends a response* $\gamma \leftarrow \mathsf{Prove}_2(\mathsf{crs},\mathsf{X},\mathsf{W},(\alpha,\beta,\mathsf{st}))$ *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* $\mathsf{Verify}(\mathsf{crs},\mathsf{X},(\alpha,\beta,\gamma))$ *and outputs* $\top$ *for acceptance and* $\bot$ *for rejection.*

The transcript (α,β,γ) *is called a* valid transcript *if* $\mathsf{Verify}(\mathsf{crs},\mathsf{X},(\alpha,\beta,\gamma)) = \top$.

[12] We define Σ-protocols in the CRS model for generality but emphasize that our concrete resulting Σ-protocols do not require them.

Below, for simplicity, we may refer to the above simply as a Σ-protocol when the meaning is clear. We require a Σ-protocol to satisfy several properties. The first property is correctness.

Definition 2.2 (Correctness). *A Σ-protocol has correctness error (δ_0, δ_1) if for any $\mathsf{crs} \in \mathsf{Setup}(1^\kappa)$ and $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$, the following holds:*

- *We have $\Pr[\mathsf{Verify}(\mathsf{crs}, \mathsf{X}, (\alpha, \beta, \gamma)) = \top] \geq 1 - \delta_0$, where the probability is take over the randomness to sample $(\alpha, \mathsf{st}) \leftarrow \mathsf{Prove}_1(\mathsf{crs}, \mathsf{X}, \mathsf{W})$, $\beta \leftarrow \mathsf{ChSet}$, and $\gamma \leftarrow \mathsf{Prove}_2(\mathsf{crs}, \mathsf{X}, \mathsf{W}, (\alpha, \beta, \gamma))$ conditioning on $\gamma \neq \bot$.*
- *The probability that an honestly generated transcript (α, β, γ) contains $\gamma = \bot$ is bounded by δ_1. In particular, $\Pr[\gamma = \bot] \leq \delta_1$ where the probability is taken over the random coins of the prover and verifier.*

We define no-abort honest-verifier zero-knowledge, a weaker variant of the standard honest-verifier zero-knowledge. For this variant, we only require the transcript to be simulatable with only knowledge of X conditioned on $\gamma \neq \bot$.

Definition 2.3 (No-abort honest-verifier zero-knowledge). *Let $D^{\gamma}_{\mathsf{trans}}(\mathsf{crs}, \mathsf{X}, \mathsf{W})$ be the distribution of $\mathsf{trans} = (\alpha, \beta, \gamma)$ from an honest protocol with prover input $(\mathsf{crs}, \mathsf{X}, \mathsf{W})$ conditioned on $\gamma \neq \bot$. Then, we say a Σ-protocol is (quantum) ϵ_{zk}-no-abort honest-verifier zero-knowledge (naHVZK), if there exists a PPT algorithm ZKSim[13] such that for all $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}$ and QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{naHVZK}}(\mathcal{A})$ defined below is less than ϵ_{zk}:*

$$\mathsf{Adv}^{\mathsf{naHVZK}}(\mathcal{A}) := \Big| \Pr[\mathsf{trans} \leftarrow D^{\gamma}_{\mathsf{trans}}(\mathsf{crs}, \mathsf{X}, \mathsf{W}) : \mathcal{A}(\mathsf{crs}, \mathsf{trans}) \to 1] - \Pr[\beta \leftarrow \mathsf{ChSet}, (\alpha, \gamma) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, \mathsf{X}, \beta) : \mathcal{A}(\mathsf{crs}, (\alpha, \beta, \gamma)) \to 1] \Big|,$$

where the probability is taken also over the randomness of $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$.

Definition 2.4 (Relaxed k-special soundness). *A Σ-protocol has* relaxed k-special soundness *if there is a deterministic PT algorithm $\mathsf{Extract}_{\mathsf{ss}}$ such that given k valid transcripts $(\alpha, \{\beta_i, \gamma_i\}_{i \in [k]})$ for any $\mathsf{crs} \in \mathsf{Setup}(1^\kappa)$ and statement $\mathsf{X} \in \mathcal{L}$ with pairwise distinct β_i's, it outputs a witness W such that $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}'$. In case $\mathcal{R}' = \mathcal{R}$, we call it "exact" special sound or simply special sound.*

The following is a stronger variant of the standard proof of knowledge that allows the knowledge extractor to directly extract from the proof output by an adversary $\mathcal{A}$ without rewinding. In the definition, the runtime of the extractor is independent of the runtime of $\mathcal{A}$ and only depends on the advantage of $\mathcal{A}$.

Definition 2.5 (Straight-line proof of knowledge). *A Σ-protocol has a (quantum) ϵ_{IndO}-*straight-line proof of knowledge ($\mathsf{SL\text{-}PoK}$) *if there exists a PPT simulator $\mathsf{SimSetup}$ and a PPT straight-line extractor $\mathsf{SL\text{-}Extract}$ with the following properties:*

- *For any QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{IndCRS}}(\mathcal{A})$ defined below is less than $\epsilon_{\mathsf{IndCRS}}$:*

$$\mathsf{Adv}^{\mathsf{IndCRS}}(\mathcal{A}) := \Big| \Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) : \mathcal{A}(1^\kappa, \mathsf{crs}) \to 1] - \Pr[(\widetilde{\mathsf{crs}}, \tau) \leftarrow \mathsf{SimSetup}(1^\kappa) : \mathcal{A}(1^\kappa, \widetilde{\mathsf{crs}}) \to 1] \Big|.$$

- *For any QPT $\mathcal{A}$ and any $\mathsf{X} \in \mathcal{L}$ satisfying*

$$\Pr\left[\begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) \\ (\alpha, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{crs}, \mathsf{X}) \\ \beta \leftarrow \mathsf{ChSet} \\ \gamma \leftarrow \mathcal{A}(\mathsf{crs}, \mathsf{X}, \alpha, \beta, \mathsf{st}) \end{array} : \mathsf{Verify}(\mathsf{crs}, \mathsf{X}, (\alpha, \beta, \gamma)) = \top \right] \geq \epsilon,$$

[13] Although we define it to be PPT for simplicity, we can consider it to be QPT. Note that this comment holds for $\mathsf{SimOracle}$ and $\mathsf{SL\text{-}Extract}$ defined in Definition 2.5.

we have

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}},\tau)\leftarrow\mathsf{SimSetup}(1^\kappa)\\(\alpha,\mathsf{st})\leftarrow\mathcal{A}(\widetilde{\mathsf{crs}},\mathsf{X})\\\beta\leftarrow\mathsf{ChSet}\\\gamma\leftarrow\mathcal{A}(\widetilde{\mathsf{crs}},\mathsf{X},\alpha,\beta,\mathsf{st})\end{array}:\begin{array}{c}\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top\\\mathsf{W}\leftarrow\mathsf{SL\text{-}Extract}(\tau,(\alpha,\beta,\gamma))\\(\mathsf{X},\mathsf{W})\in\mathcal{R}'\end{array}\right]\geq\frac{\epsilon-\nu_1}{p_1},$$

for some polynomial p_1 and negligible function ν_1. Moreover, the runtime of SL-Extract *is upper bounded by $p_2\cdot\left(\frac{\epsilon-\nu_2}{p_3}-\frac{1}{|\mathsf{ChSet}|}\right)^{-1}$ for some polynomials p_2,p_3 and negligible function ν_2.*[14] *Concretely, if ϵ is non-negligible and $|\mathsf{ChSet}|$ is super-polynomially large, then* SL-Extract *runs in polynomial time.*

Finally, we provide one additional property of Σ-protocol we require for the Fiat-Shamir transform.

Definition 2.6 (Min-entropy). *A Σ-protocol has ζ-min-entropy if for all $(\mathsf{X},\mathsf{W})\in\mathcal{R}$, and (possibly unbounded) quantum algorithm $\mathcal{A}$, we have*

$$\Pr[\mathsf{crs}\leftarrow\mathsf{Setup}(1^\kappa),(\alpha,\mathsf{st})\leftarrow\mathsf{Prove}_1(\mathsf{crs},\mathsf{X},\mathsf{W}),\alpha'\leftarrow\mathcal{A}(\mathsf{crs}):\alpha=\alpha']\leq 2^{-\zeta}.$$

2.2 Quantum Background

Quantum Computation. We briefly give some backgrounds 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 $|x\rangle$ to $U|x\rangle$. 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 $\mathcal{A}$. (An oracle query is counted as a unit time if $\mathcal{A}$ is an oracle algorithm.) Any efficient classical computation can be realized by a quantum computation efficiently. That is, for any function f that is classically computable, there exists a unitary matrix U_f such that $U_f|x,y\rangle=|x,f(x)\oplus y\rangle$, and the number of universal gates to express U_f is linear in the size of a classical circuit that computes f.

Quantum random oracle model. Boneh et al. [BDF+11] introduced the quantum random oracle model (QROM), which is an extension of the usual random oracle model to the quantum setting. Roughly speaking, the QROM is an idealized model where a hash function is idealized to be a quantumly accessible oracle that simulates a random function. More precisely, in security proofs in the QROM, a random function $\mathsf{H}:\mathcal{X}\to\mathcal{Y}$ is uniformly chosen at the beginning of the experiment, and every entity involved in the system is allowed to access the oracle H, which on input $\sum_{x,y}\alpha_{x,y}|x,y\rangle$ returns $\sum_{x,y}\alpha_{x,y}|x,\mathsf{H}(x)\oplus y\rangle$. We denote a quantum algorithm $\mathcal{A}$ that accesses the oracle H by $\mathcal{A}^{|\mathsf{H}\rangle}$. In the QROM, one query to the random oracle is counted as one unit time. Although we do not require it, recently, Zhandry [Zha19] devised a new proof technique for simulating the QRO without having to commit to one fixed description of a random function at the beginning of the security game. One of the benefits of using the prior proof technique is that the proof follows much like the classical counterpart and requires minimal background on quantum computation.

Useful lemmas. We prepare a minimal set of lemmas regarding quantum computation.

Definition 2.7 (Quantum-accessible PRF). *We say that a function $\mathsf{PRF}:\mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ is a* quantum-accessible pseudorandom function (PRF) *if for all QPT adversaries $\mathcal{A}$, its advantage defined below is negligible:*

$$\mathsf{Adv}^{\mathsf{qaPRF}}(\mathcal{A}):=\left|\Pr[\mathcal{A}^{|\mathsf{RF}\rangle}(1^\kappa)\to 1]-\Pr[\mathcal{A}^{|\mathsf{PRF}(\mathsf{K},\cdot)\rangle}(1^\kappa)\to 1]\right|,$$

where $\mathsf{RF}\leftarrow\mathsf{Func}(\mathcal{X},\mathcal{Y})$ and $\mathsf{K}\leftarrow\mathcal{K}$.

[14] In case the term inside $(\cdot)^{-1}$ is a non-positive, it is understood that SL-Extract simply outputs $\perp$ on invocation.

Zhandry [Zha12a] proved that some known constructions of classical PRFs including the classical construction of [GGM86] and the lattice-based construction of [BPR12] are also quantum-accessible PRFs. Moreover, it is known that as in the classical ROM, we can use a QRO as a PRF [SXY18].

Lemma 2.8 ([Zha12a, Theorem 1.1]). *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*

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

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 $\epsilon^2/(C \cdot Q^3)$ for some universal constant $C > 0$.

Above, by removing the requirements that D_0 and D_1 are efficiently sampleable and $\mathcal{A}$ runs in polynomial time, we obtain a statistical variant of the lemma. Below, 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 2.9 (Generic search problem with bounded probabilities, [KLS18, Lemma 2.1]). *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 $\mathsf{Adv}^{\mathsf{GSBP}}(\mathcal{A}) := \Pr[\mathcal{A}\ wins] \leq 8 \cdot \lambda \cdot (Q+1)^2$.

2.3 Lattices

Rings and Gaussian Measures. For positive integers d and q, let R_q denote the polynomial ring $\mathbb{Z}_q[X]/(X^d+1)$. Throughout this paper we view ring elements in $a = \sum_{i=0}^{d-1} \alpha_i X^i \in \mathbb{Z}[X]/(X^d+1)$ as a vector $(\alpha_0, \cdots, \alpha_{d-1})^\top \in \mathbb{Z}^d$ interchangeably. For a positive real σ, let $D_{\mathbb{Z}^d,\sigma}$ denote the discrete Gaussian distribution over $\mathbb{Z}^d$. To make the notation simple, we denote $a \leftarrow D_\sigma$ to mean that the coefficient vectors of $a \in R_q$ is sampled from $D_{\mathbb{Z}^d,\sigma}$. The definition naturally extends to vectors $\boldsymbol{a} \in R^m$ by viewing $\boldsymbol{a}$ as a vector in $\mathbb{Z}_q^{md}$. Finally, let S_η denote the set of all elements in $a \in R_q$ such that $\|w\|_\infty \leq \eta$. The followings are some useful tools regarding Gaussian distributions.

Lemma 2.10 (Rejection Sampling, [Lyu12, Lemmas 4.3, 4.6]). *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 Figure 2. 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 $h \times D_{\mathbb{Z}^m,\sigma}$.*

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 is a useful lemma to bound the norm of an element sampled from $D_{\mathbb{Z}^n,\sigma}$.

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

1: $u \leftarrow [0, 1)$

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 2: Rejection sampling.

Lemma 2.11 ([MR04, Lyu12]). *For any real $t > 0$ and $t' > 1$, we have*

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

The MLWE assumption. We define a variant of the standard module learning with errors MLWE assumption (which remains as hard as the standard MLWE assumption), where the adversary is allowed to obtain a superposition of *independent* MLWE samples.

Definition 2.12 (Quantum accessible MLWE). *For integers $n = n(\kappa), m = m(n), q = q(n) > 2, L = L(\kappa)$, an error distribution $\chi = \chi(n)$ over R_q, and a QPT algorithm $\mathcal{A}$ that makes at most Q oracle queries, the advantage of the* quantum accessible *module learning with errors $\mathsf{qaMLWE}_{n,m,L,Q,\chi}$ problem of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{qaMLWE}_{n,m,L,Q,\chi}}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{|\mathcal{O}_{\mathsf{LWE}}\rangle}(1^\kappa) \to 1] - \Pr[\mathcal{A}^{|\mathcal{O}_{\$}\rangle}(1^\kappa) \to 1]\right|,$$

where oracles $\mathcal{O}_{\mathsf{MLWE}}$ and $\mathcal{O}_{\$}$ are defined as

- $\mathcal{O}_{\mathsf{MLWE}}$: *On input $t \in [L]$, sample $\mathbf{A} \leftarrow R_q^{m \times n}$, $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow \chi^n \times \chi^m$, and output $(\mathbf{A}, \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$;*

- $\mathcal{O}_{\$}$: *On input $t \in [L]$, sample $(\mathbf{A}, \mathbf{b}) \leftarrow R_q^{m \times n} \times R_q^m$ and output $(\mathbf{A}, \mathbf{b})$.*

We assume the oracles run on a uniform and independent randomness for each input $t \in [L]$ and reuse the same randomness when run again on the same t.

When $L = 1$, qaMLWE is equivalent to the standard MLWE [Reg09, LS15] and we simply call it $\mathsf{MLWE}_{n,m,\chi}$. When $L > 1$, since all the secrets and noises are sampled independently for each $t \in [L]$, a naive hybrid argument shows that qaMLWE is equivalent to the standard MLWE with reduction loss $1/L$. Moreover, using Lemma 2.8, an adversary $\mathcal{A}$ making Q oracle queries against qaMLWE with advantage ϵ can be used to construct an adversary $\mathcal{B}$ against the standard MLWE with advantage $\epsilon^2/(C \cdot Q^3)$, where the reduction loss is independent of L. This is useful in scenarios when L is exponentially large. When we consider qaMLWE where the error distribution χ is uniform random over the set S_η, we simply write $\mathsf{qaMLWE}_{n,m,L,Q,\eta}$. We use the same convention for the following hardness assumption.

We note that [GKZ19] considers a setting where an adversary can obtain a superposition of LWE samples for *all* $\mathbf{A} \in R_q^{m \times n}$ with the *same* secret, i.e., $\sum_{\mathbf{A} \in R_q^{m \times n}} |\mathbf{A}\rangle |\mathbf{A} \cdot \mathbf{s} + \mathbf{e}_{\mathbf{A}}\rangle$. They showed that such a problem can be broken in quantum polynomial time. However, such an attack cannot be applied to our setting since in our setting, the output of $\mathcal{O}_{\mathsf{MLWE}}$ for each input are identically and independently distributed.

The DSMR assumption. We define the decisional small *matrix* ratio (DSMR) assumption that generalizes the decisional small *polynomial* ratio (DSPR) assumption used by [LTV12, SXY18]. While the DSPR problem is a hardness assumption defined over the so-called NTRU lattices [HPS98], the DSMR problem is defined over a *module* NTRU lattice [**?**]. Similarly to above, we consider the quantum accessible variant which is as secure as the standard DSPR by either applying a naive hybrid argument or Lemma 2.8.

Definition 2.13 (Quantum accessible DSMR). *For integers $n = n(\kappa), m = m(n), p = p(n), q = q(n) > 2, L = L(\kappa)$ where $p < q$ are coprime odd integers, a distribution $\chi = \chi(n)$ over R_q, and a QPT algorithm $\mathcal{A}$*

that makes at most Q oracle queries, the advantage of the quantum accessible *decisional small matrix ratio* $\mathsf{qaDSMR}_{n,m,L,\chi}$ *problem of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{qaDSMR}_{n,m,L,Q,\chi}}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{|\mathcal{O}_{\mathsf{DSMR}}\rangle}(1^\kappa) \to 1] - \Pr[\mathcal{A}^{|\mathcal{O}_{\$}\rangle}(1^\kappa) \to 1]\right|,$$

where oracles $\mathcal{O}_{\mathsf{DSMR}}$ and $\mathcal{O}_{\$}$ are defined as

- $\mathcal{O}_{\mathsf{DSMR}}$: *On input $t \in [L]$, sample $(\mathbf{F}, \mathbf{G}) \leftarrow \chi^{m\times m} \times \chi^{m\times n}$ conditioned on $\mathbf{F}$ being invertible over* mod q *and* mod p*, and output $\mathbf{H} = p \cdot \mathbf{F}^{-1}\mathbf{G} \in R_q^{m\times n}$*

- $\mathcal{O}_{\$}$: *On input $t \in [L]$, sample $\mathbf{H} \leftarrow R_q^{m\times n}$ and output $\mathbf{H}$.*

We assume the oracles run on a uniform and independent randomness for each input $t \in [L]$ and reuse the same randomness when run again on the same t.

3 Extractable Linear Homomorphic Commitment Protocol

In this section, we introduce a new interactive protocol called the *extractable linear homomorphic commitment* (LinHC) protocol. We first provide the definition of an extractable LinHC protocol and then give two instantiations: one from the MLWE assumption and the other from the MLWE *and* the DSMR assumption. Below whenever we say Σ-protocols, the readers may safely replace them by public-coin HVZK non-interactive protocols.

We first define extractable LinHC protocol in its most general form and provide a simplified variant in the subsequent section. As explained in the introduction, the general definition, which is defined in the QROM, is useful when directly constructing (straight-line simulation extractable) NIZKs[15] in the QROM from a possibly non-quantum secure Σ-protocol (see Section 4.2). In contrast, the simplified definition, which is defined in the standard model, is useful when constructing a quantum straight-line proof of knowledge Σ-protocol from a non-quantum secure Σ-protocol (see Section 4.1).

3.1 Definition

An illustration of the extractable LinHC protocol is provided in Figure 3. Looking ahead, in the context of Σ-protocols, the $\mathbf{e}_i$'s and $\mathbf{r}$ correspond to the witness and commitment randomness (or masking term), respectively.

Prover: $(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$		Verifier: $\mathsf{K}_{\mathsf{com}}$
$(\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\boldsymbol{\beta} = (\beta_1, \cdots, \beta_N)}$	$\boldsymbol{\beta} \leftarrow \mathsf{ChSet}$
$\mathbf{z} = \sum_{i=1}^N \beta_i \cdot \mathbf{e}_i + \mathbf{r}$ $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \mathbf{z}), \mathsf{st})$	$\xrightarrow{(\mathbf{z}, \mathsf{op})}$	
		$\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))) \stackrel{?}{=} \top$

Figure 3: An overview of an extractable linear homomorphic commitment protocol. $\mathsf{K}_{\mathsf{com}}$ is a commitment key generated by $\mathsf{KeyGen}^{\mathsf{H}}(1^\kappa)$, where H is modeled as a (quantum) random oracle.

Definition 3.1 (Extractable linear homomorphic commitment protocol in QROM). *An* extractable linear homomorphic commitment (LinHC) protocol *is a three-round public-coin interactive protocol run between two parties (prover and verifier), and is defined by a tuple of PPT algorithms* $\Pi_{\mathsf{LinHC}} = (\mathsf{KeyGen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Verify})$ *and a challenge set* $\mathsf{ChSet} \subseteq (R_q)^N$*. The protocol procedure is as follows:*

[15] Roughly, this is type of NIZK that, even after seeing many simulated proofs, whenever an adversary outputs a valid proof, we can straight-line extract a witness from the proof [FKMV12].

1. *A random oracle* H *is chosen and the* key generation *algorithm is executed* $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}}(1^{\kappa})$. *Here, let* $\{0,1\}^{\nu}$ *be the randomness space used by* KeyGen*;*

2. *The prover on input vectors* $((\mathbf{e}_i)_{i\in[N]}, \mathbf{r}) \in (R_q^m)^N \times R_q^m$*, runs the* commitment *algorithm* $(\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$*, and sends the first message* com *to the verifier;*

3. *The verifier samples a random challenge* $\boldsymbol{\beta} \leftarrow \mathsf{ChSet}$ *and sends the second message* $\boldsymbol{\beta}$ *to the prover;*

4. *The prover computes* $\mathbf{z} \leftarrow \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}$[16]*, runs the* opening *algorithm* $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \mathbf{z}), \mathsf{st})$*, and sends the third message* $(\mathbf{z}, \mathsf{op})$ *to the verifier. We allow* $\mathsf{op} = \perp$ *for a special symbol* $\perp$ *to indicate failure;*

5. *The verifier returns the output of the deterministic* verification *algorithm* $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))$*, where* $\top$ *indicates accept and* $\perp$ *indicates reject. We call* $(\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ *the* transcript *and call* $(\mathsf{com}, \boldsymbol{\beta}, \mathsf{op})$ *a* valid opening *for* $\mathbf{z}$ *if the verifier accepts.*

We require the following properties to hold.

Definition 3.2 (Correctness). *An extractable linear homomorphic commitment protocol* Π_{LinHC} *has correctness error* (δ_0, δ_1) *if for any choice of random oracle* H*,* $\mathsf{K}_{\mathsf{com}} \in \mathsf{KeyGen}^{\mathsf{H}}(1^{\kappa})$*, and* $((\mathbf{e}_i)_{i\in[N]}, \mathbf{r}) \in (R_q^m)^N \times R_q^m$ *the following holds:*

- *We have* $\Pr[\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))) = \top] \geq 1 - \delta_1$*, where the probability is taken over the randomness to sample* $(\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$*,* $\boldsymbol{\beta} \leftarrow \mathsf{ChSet}$*, and* $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}), \mathsf{st})$ *conditioned on* $\mathsf{op} \neq \perp$*.*

- *The probability that an honestly generated transcript* $(\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ *contains* $\mathsf{op} = \perp$ *is bounded by* δ_1*. In particular,* $\Pr[\mathsf{op} = \perp] \leq \delta_1$ *where the probability is taken over the random coins of the prover and verifier.*

Zero-knowledge. At a high level, zero-knowledge for an extractable LinHC protocol stipulates that the transcript should leak no information of the vectors $(\mathbf{e}_i)_{i\in[N]}$ and $\mathbf{r}$ other than the fact that it adds up to $\mathbf{z}$. Below, we provide a definition of zero-knowledge where an adversary can obtain superpositions of simulated proofs. Since $(\mathbf{e}_i)_{i\in[N]}$ corresponds to the witness of the underlying Σ-protocol, it will be reused many times. On the other hand, $\mathbf{r}$ is the commitment randomness that is freshly sampled for each trasncript. This is reflected in the following definition by fixing $(\mathbf{e}_i)_{i\in[N]}$ and sampling fresh $\mathbf{r}$ (and challenge $\boldsymbol{\beta}$) using the distribution $D_{\boldsymbol{\beta},\mathbf{r}}$. Also, one can think of each ρ in the definition as a specific tag to distinguish each transcripts. Below, we say it is "semi"-honest-verifier since $\boldsymbol{\beta}$ does not necessarily need to be uniformly distributed over ChSet.

Definition 3.3 (Quantum accessible no-abort (semi-)honest-verifier zero-knowledge). *Let* $D_{\boldsymbol{\beta},\mathbf{r}}$ *be any distribution over* $\mathsf{ChSet} \times R_q^m$*. For an oracle* H *and algorithm* ZKSim*, define the following algorithms:*

- $D_{\mathsf{trans}}^{\mathcal{V}}(\rho, (\mathbf{e}_i)_{i\in[N]})$: *On input* $\rho \in \{0,1\}^{\nu}$ *and* $(\mathbf{e}_i)_{i\in[N]} \in (R_q^m)^N$*, generate* $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}}(1^{\kappa})[\rho]$ *and sample* $(\boldsymbol{\beta}, \mathbf{r}) \leftarrow D_{\boldsymbol{\beta},\mathbf{r}}$*. Then run an honest protocol with prover input* $(\mathsf{K}_{\mathsf{com}}, ((\mathbf{e}_i)_{i\in[N]}, \mathbf{r}))$ *conditioned on the verifier message being* $\boldsymbol{\beta}$ *and* $\mathsf{op} \neq \perp$ *(i.e., a non-aborting protocol). Finally, output* $\mathbf{r}$ *along with the valid transcript* $(\mathbf{r}, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))$*.*

- $D_{\mathsf{sim}}(\rho, (\mathbf{e}_i)_{i\in[N]})$: *On input* $\rho \in \{0,1\}^{\nu}$ *and* $(\mathbf{e}_i)_{i\in[N]} \in (R_q^m)^N$*, generate* $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}}(1^{\kappa})[\rho]$*, sample* $(\boldsymbol{\beta}, \mathbf{r}) \leftarrow D_{\boldsymbol{\beta},\mathbf{r}}$*, and compute* $\mathbf{z} \leftarrow \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}$*. Then, run* $(\mathsf{com}, \mathsf{op}) \leftarrow \mathsf{ZKSim}(\mathsf{K}_{\mathsf{com}}, \boldsymbol{\beta}, \mathbf{z})$ *and output* $(\mathbf{r}, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))$*.*

[16] Although it suffices to consider $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$ in many cases, there are recent protocols that require this extra level of generality, e.g., [ESLL19].

In above, we assume $D^{\chi}_{\mathsf{trans}}$ and D_{sim} run on a uniform and independent randomness for each input $\rho \in \{0,1\}^{\nu}$ and reuse the same randomness when run again on the same ρ.

*Then, we say an extractable linear homomorphic commitment protocol Π_{LinHC} has ϵ_{zk}-*quantum accessible no-abort (semi-)honest-verifier zero-knowledge *(*$\mathsf{QAnaHVZK}$*), if there exists a PPT algorithm* ZKSim *such that for any $(\mathbf{e}_i)_{i\in[N]} \in (R_q^m)^N$, distribution $D_{\boldsymbol{\beta},\mathbf{r}}$, and QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{QAnaHVZK}}(\mathcal{A})$ defined below is less than ϵ_{zk}:*

$$\mathsf{Adv}^{\mathsf{QAnaHVZK}}(\mathcal{A}) := \left| \Pr\left[\mathcal{A}^{|\mathsf{H}\rangle,|D^{\chi}_{\mathsf{trans}}(\cdot,(\mathbf{e}_i)_{i\in[N]})\rangle}(1^{\kappa}) \to 1\right] - \Pr\left[\mathcal{A}^{|\mathsf{H}\rangle,|D_{\mathsf{sim}}(\cdot,(\mathbf{e}_i)_{i\in[N]})\rangle}(1^{\kappa}) \to 1\right] \right|,$$

where the probability is also taken over the random choice of the random oracle H.

Extractability. When considering extractable LinHC protocol as a tool to be integrated into a preexisting Σ-protocol, the third message $\mathbf{z}$ corresponds to the third message (usually referred to as the "response") of the Σ-protocol. See Figure 8 for an illustrative example. In particular, the verifier will always perform an additional check $f(\boldsymbol{\beta},\mathbf{z}) \stackrel{?}{=} \top$, where f is some function defined by the verifier algorithm of the underlying Σ-protocol. Therefore, for an extractable LinHC to be useful in the context of Σ protocols, we want it to be able to extract valid tuples $\{(\boldsymbol{\beta}_i,\mathbf{z}_i)\}_{i\in[k]}$ such that $f(\boldsymbol{\beta}_i,\mathbf{z}_i) = \top$ *without* rewinding the adversary only given an accepting transcript. After such k tuples are collected, we can invoke the k-special soundness extractor of the underlying Σ-protocol to extract a witness. More formally, we require the following.

Definition 3.4 ($\mathcal{F}$-Almost straight-line extractable). *Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output space required by the random oracle* H. *An extractable linear homomorphic commitment protocol Π_{LinHC} is ϵ_{IndO}-$\mathcal{F}$-*almost straight-line extractable *for a function family $\mathcal{F}$ if there exists PPT algorithms* $\mathsf{SimOracle}$ *and* $\mathsf{LinCExtract}$ *with the following properties:*

1. *For any QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{IndO}}(\mathcal{A})$ defined below is less than ϵ_{IndO}:*

$$\mathsf{Adv}^{\mathsf{IndO}}(\mathcal{A}) := \left| \Pr[\mathsf{H} \leftarrow \mathsf{Func}(\mathcal{X},\mathcal{Y}) : \mathcal{A}^{|\mathsf{H}\rangle}(1^{\kappa}) \to 1] - \Pr[(\widetilde{\mathsf{H}},\tau) \leftarrow \mathsf{SimOracle}(1^{\kappa}) : \mathcal{A}^{|\widetilde{\mathsf{H}}\rangle}(1^{\kappa}) \to 1] \right|.$$

2. *For any $(\widetilde{\mathsf{H}},\tau) \in \mathsf{SimOracle}(1^{\kappa})$, randomness $\rho \in \{0,1\}^{\nu}$, first message* com, *and any efficiently computable function $f \in \mathcal{F}$ with binary output $\{\top,\bot\}$, define the set*

$$S_f(\rho,\mathsf{com}) := \{\boldsymbol{\beta} \mid \exists(\mathbf{z},\mathsf{op}) \text{ s.t. } \mathsf{Verify}(\mathsf{K}_{\mathsf{com}},(\mathsf{com},\boldsymbol{\beta},(\mathbf{z},\mathsf{op}))) = \top \ \wedge \ f(\boldsymbol{\beta},\mathbf{z}) = \top\},$$

where $\mathsf{K}_{\mathsf{com}} = \mathsf{KeyGen}^{\widetilde{\mathsf{H}}}(1^{\kappa})[\rho]$. Let δ,k be any positive integers such that $k < |S_f(\rho,\mathsf{com})|$, and denote $T^* = \frac{k\cdot\delta\cdot|\mathsf{ChSet}|}{|S_f(\rho,\mathsf{com})|-k}$. Then, on input a valid transcript $\mathsf{trans} = (\mathsf{com},\boldsymbol{\beta},(\mathbf{z},\mathsf{op}))$, the linear commitment extractor $\mathsf{LinCExtract}(\tau,\rho,\mathsf{trans})$ *outputs either a set $L = \{(\boldsymbol{\beta}_j,\mathbf{z}_j)\}_{j\in[k]}$ or $\bot$ in time $T^* \cdot \mathsf{poly}(\kappa)$ for some fixed polynomial $\mathsf{poly}(\kappa)$, where all the $\boldsymbol{\beta}_j$'s in L are pairwise distinct and satisfies $f(\boldsymbol{\beta}_j,\mathbf{z}_j) = \top$. Moreover, the probability that it outputs L is at least $1 - k\cdot 2^{-\delta}$. Concretely, when k is a constant, $\delta = \kappa$, and $|S_f(\rho,\mathsf{com})| = |\mathsf{ChSet}|\cdot\epsilon$ for a non-negligible ϵ, then* $\mathsf{LinCExtract}$ *outputs L in polynomial time with overwhelming probability.*

In general we cannot efficiently check if the extracted $\boldsymbol{\beta}_j$ satisfies $\boldsymbol{\beta}_j \in S_f(\rho,\mathsf{com})$ since we cannot extract op_j corresponding to $(\boldsymbol{\beta}_j,\mathbf{z}_j)$, hence the term "almost" straight-line extractable. This implies that the set L may include an invalid $(\boldsymbol{\beta}_j,\mathbf{z}_j)$ for which there does not exist a valid op_j. However, this will not be an issue for most of our application where f defines the entire verification algorithm of the underlying Σ-protocol. In these cases, we only need $f(\boldsymbol{\beta}_j,\mathbf{z}_j) = \top$ for k-tuples to hold to invoke the k-special soundness extractor. We also point out that in many cases we are not able to efficiently compute the cardinality of the set $S_f(\rho,\mathsf{com})$ so we do not know if $\mathsf{LinCExtract}$ runs in polynomial time. However, in typical applications, we can deduce that $S_f(\rho,\mathsf{com})$ must be of size $|\mathsf{ChSet}|\cdot\epsilon$ for a non-negligible ϵ unless the adversary breaks some other intractable problem.

Optional. Finally, we consider two optional properties for $\mathcal{F}$-almost straight-line extractability. The following is useful in situations where the function f does not comprise the entire verification algorithm of the underlying Σ-protocol. In these situations, collecting $(\boldsymbol{\beta}_j, \mathbf{z}_j)_{j\in[k]}$ may not suffice to invoke the special soundness extractor.

Definition 3.5 (Optional properties). *The definition of $\mathcal{F}$-almost straight-line extractability in Definition 3.4 can be augmented by the following two optional properties:*

3. (small challenge set) *In case* ChSet *is only of polynomial size, then* $\mathsf{LinCExtract}(\tau, \rho, \mathsf{trans})$ *outputs a set* $L = \{(\boldsymbol{\beta}_j, \mathbf{z}_j)\}_{j\in[M]}$ *in time* $|\mathsf{ChSet}| \cdot \mathsf{poly}(\kappa)$ *for some* M *and fixed polynomial* $\mathsf{poly}(\kappa)$ *where all the* $\boldsymbol{\beta}_j$*'s in* L *are pairwise distinct and there exists* $|S_f(\rho, \mathsf{com})|$*-tuples* $(\boldsymbol{\beta}_j, \mathbf{z}_j)$ *in* L *such that* $\boldsymbol{\beta}_j \in S_f(\rho, \mathsf{com})$ *and* $f(\boldsymbol{\beta}_j, \mathbf{z}_j) = \top$*.*

4. (uniqueness) *For any* $(\widetilde{\mathsf{H}}, \tau) \in \mathsf{SimOracle}(1^\kappa)$*, randomness* ρ*, and first message* com*, there exists at most one* $\mathbf{z}$ *for each* $\boldsymbol{\beta} \in \mathsf{ChSet}$ *such that there exists a valid* op *satisfying* $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, \mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})) = \top$*, where* $\mathsf{K}_{\mathsf{com}} = \mathsf{KeyGen}^{\widetilde{\mathsf{H}}}(1^\kappa)[\rho]$*.*

Remark 3.6 (Classical definition). All of the above definitions can be turned into a classical definition by replacing the QPT algorithms to classical PPT algorithms. We believe the classical case can be of an independent interest since it can be used as an alternative to the Fischlin transform [Fis05] that realizes straight-line extractable NIZKs in the classical ROM.

3.2 Simplified Definition of Extractable LinHC

As explained earlier, in case the goal is to construct quantum secure Σ-protocols (and not a QROM secure simulation extractable NIZK or a signature), we can use a simplified definition of extractable LinHC protocols in the standard model. One of the main simplification comes from the fact that since all of the security notions are decoupled from the QRO, the proofs follow much like the classical counterparts. For example, zero-knowledge of a simplified extractable LinHC protocol is defined similarly to standard naHVZK of a Σ-protocol. Details follows.

Definition 3.7 (Simplified extractable linear homomorphic commitment protocol). *A simplified version of the extractable linear homomorphic commitment protocol is defined exactly as in Definition 3.1 except that a random oracle* H *is no longer sampled and the key generation algorithm* KeyGen *is executed without having access to any oracle.*

Correctness is defined exactly as in Definition 3.2 except that we get rid of the random oracle H. We only require a simple definition of naHVZK defined as follows:

Definition 3.8 (Simplified no-abort honest-verifier zero-knowledge). *Let* $\mathsf{W} = ((\mathbf{e}_i)_{i\in[N]}, \mathbf{r}) \in (R_q^m)^N \times R_q^m$ *and* $D^{\chi}_{\mathsf{trans}}(\mathsf{K}_{\mathsf{com}}, \mathsf{W})$ *be the distribution of* $\mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ *from an honest protocol with prover input* $(\mathsf{K}_{\mathsf{com}}, \mathsf{W})$ *conditioned on* $\mathsf{op} \neq \perp$*. Then, we say a simplified extractable linear homomorphic commitment protocol* Π_{LinHC} *has (quantum)* ϵ_{zk}-no-abort honest-verifier zero-knowledge (naHVZK)*, if there exists a PPT algorithm* ZKSim *such that for all* W *and QPT* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{\mathsf{naHVZK}}(\mathcal{A})$ *defined below is less than* ϵ_{zk}*:*

$$\mathsf{Adv}^{\mathsf{naHVZK}}(\mathcal{A}) := \Big| \Pr\left[\mathsf{trans} \leftarrow D^{\chi}_{\mathsf{trans}}(\mathsf{K}_{\mathsf{com}}, \mathsf{W}) \ : \mathcal{A}(\mathsf{K}_{\mathsf{com}}, \mathsf{W}, \mathsf{trans}) \to 1 \right] - \Pr\left[\begin{array}{c} \boldsymbol{\beta} \leftarrow \mathsf{ChSet},\ \mathbf{z} \leftarrow \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}, \\ (\mathsf{com}, \boldsymbol{\beta}, \mathsf{op}) \leftarrow \mathsf{ZKSim}(\mathsf{K}_{\mathsf{com}}, \mathbf{z}) \end{array} : \mathcal{A}(\mathsf{K}_{\mathsf{com}}, \mathsf{W}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))) \to 1 \right] \Big|,$$

where the probability is also taken over the random choice of $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}(1^\kappa)$*.*

Finally, we define a slightly simplified definition of $\mathcal{F}$-almost straight-line extractable along with its optional requirements.

Definition 3.9 (Simplified $\mathcal{F}$-Almost straight-line extractable). *A simplified extractable linear homomorphic commitment protocol Π_{LinHC} is $\epsilon_{\mathsf{IndCom}}$-$\mathcal{F}$-*almost straight-line extractable *for a function family $\mathcal{F}$ if there exist PPT algorithms* SimKeyGen *and* LinCExtract *with the following properties:*

1. *For any QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{IndCom}}(\mathcal{A})$ defined below is less than $\epsilon_{\mathsf{IndCom}}$:*

$$\mathsf{Adv}^{\mathsf{IndCom}}(\mathcal{A}) := \left| \Pr[\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}(1^\kappa) : \mathcal{A}(1^\kappa, \mathsf{K}_{\mathsf{com}}) \to 1] - \Pr[(\widetilde{\mathsf{K}}_{\mathsf{com}}, \tau) \leftarrow \mathsf{SimKeyGen}(1^\kappa) : \mathcal{A}(1^\kappa, \widetilde{\mathsf{K}}_{\mathsf{com}}) \to 1] \right|.$$

2. *For any $(\widetilde{\mathsf{K}}_{\mathsf{com}}, \tau) \in \mathsf{SimKeyGen}(1^\kappa)$, first message* com*, and any efficiently computable function $f \in \mathcal{F}$ with binary output $\{\top, \bot\}$, define the set*

$$S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com}) := \{\boldsymbol{\beta} \mid \exists(\mathbf{z}, \mathsf{op}) \ s.t. \ \mathsf{Verify}(\widetilde{\mathsf{K}}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})) = \top \ \wedge \ f(\boldsymbol{\beta}, \mathbf{z}) = \top)\}.$$

Let δ, k be any positive integers such that $k < |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com})|$, and denote $T^* = \frac{k \cdot \delta \cdot |\mathsf{ChSet}|}{|S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com})| - k}$. Then, on input a valid transcript $\mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$, the linear commitment extractor $\mathsf{LinCExtract}(\tau, \mathsf{trans})$ outputs either a set $L = \{(\boldsymbol{\beta}_j, \mathbf{z}_j)\}_{j \in [k]}$ or $\bot$ in time $T^* \cdot \mathsf{poly}(\kappa)$ for some fixed polynomial $\mathsf{poly}(\kappa)$, where all the $\boldsymbol{\beta}_j$'s in L are pairwise distinct and satisfies $f(\widetilde{\boldsymbol{\beta}}_j, \widetilde{\mathbf{z}}_j) = \top$. Moreover, the probability that it outputs L is at least $1 - k \cdot 2^{-\delta}$. Concretely, when k is a constant, $\delta = \kappa$, and $|S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com})| = |\mathsf{ChSet}| \cdot \epsilon$ for a non-negligible ϵ, then LinCExtract *outputs L in polynomial time with overwhelming probability.*

Definition 3.10 (Optional properties). *The definition of the simplified $\mathcal{F}$-almost straight-line extractability in Definition 3.9 can be augmented by the following two optional properties:*

3. *(*small challenge set*) In case* ChSet *is only of polynomial size, then $\mathsf{LinCExtract}(\tau, \mathsf{trans})$ outputs a set $L = \{(\boldsymbol{\beta}_j, \mathbf{z}_j)\}_{j \in [M]}$ in time $|\mathsf{ChSet}| \cdot \mathsf{poly}(\kappa)$ for some M and fixed polynomial $\mathsf{poly}(\kappa)$ where all the $\boldsymbol{\beta}_j$'s in L are pairwise distinct and there exists $|S_f(\rho, \mathsf{com})|$-tuples $(\boldsymbol{\beta}_j, \mathbf{z}_j)$ in L such that $\boldsymbol{\beta}_j \in S_f(\rho, \mathsf{com})$ and $f(\boldsymbol{\beta}_j, \mathbf{z}_j) = \top$.*

4. *(*uniqueness*) For any $(\widetilde{\mathsf{K}}_{\mathsf{com}}, \tau) \in \mathsf{SimKeyGen}(1^\kappa)$, and first message* com*, there exists at most one $\mathbf{z}$ for each $\boldsymbol{\beta} \in \mathsf{ChSet}$ such that there exists a valid* op *satisfying $\mathsf{Verify}(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})) = \top$.*

3.3 Interlude: Extractable LinHC Specialized for Lattices

In most, if not all, lattice-based Σ-protocols, the witness being proven is a "short" vector. Therefore, throughout this work, we assume such shortness condition holds by default and integrate it into the definition of the extractable LinHC protocol. Effectively, we are able to construct a more efficient extractable LinHC protocol by taking advantage of these bounds.

Norm bound on $(\mathbf{e}_i)_{i \in [N]}$ and $\mathbf{r}$. In the following, we assume the size of the vectors $(\mathbf{e}_i)_{i \in [N]}$ and $\mathbf{r}$ in R_q^m have an upper bound. That is, for all $i \in [N]$, there exist positive integers $B_{\infty,\mathbf{e}}, B_{2,\mathbf{e}}, B_{\infty,\mathbf{r}}$, and $B_{2,\mathbf{r}}$ such that $\|\mathbf{e}_i\|_\infty \leq B_{\infty,\mathbf{e}_i}$, $\|\mathbf{e}_i\|_2 \leq B_{2,\mathbf{e}_i}$, $\|\mathbf{r}\|_\infty \leq B_{\infty,\mathbf{r}}$ and $\|\mathbf{r}\|_2 \leq B_{2,\mathbf{r}}$. In particular, we only guarantee correctness and naHVZK for such $\mathbf{e}_i$'s and $\mathbf{r}$.

Restricting the function class $\mathcal{F}$ to check norm bound. As explained in the previous section, the function class $\mathcal{F}$ of $\mathcal{F}$-almost straight-line extractability (Definition 3.4) corresponds to the the check performed by the verifier of the underlying Σ-protocol, which we are trying to make secure in the (Q)ROM via extractable LinHC. Namely, the verifier of the Σ-protocol receives $\mathbf{z}$ from the prover and then checks whether some condition $f \in \mathcal{F}$ holds with respect to the challenge $\boldsymbol{\beta}$ it sampled, i.e., $f(\boldsymbol{\beta}, \mathbf{z}) \stackrel{?}{=} \top$. In any lattice-based Σ-protocol, one of the conditions that is always checked by the verifier is whether $\mathbf{z}$ is "small" (see Section 4.1 for a concrete example). We therefore restrict the function class $\mathcal{F}$ to be a family of functions $\mathcal{F}_B$ such that for any $f \in \mathcal{F}_B$, f includes the check $\|\mathbf{z}\|_2 \leq B$.[17] In many lattice-based Σ-protocols, we have $B \approx B_{\infty,\mathbf{r}}$ or $B_{2,\mathbf{r}}$, where recall $\mathbf{r}$ is the "masking" term to hide $(\mathbf{e}_i)_{i \in [N]}$.

[17] The choice of the Euclidean norm is arbitral and we can also chose the infinity norm (or include both norms).

3.4 First Construction of Extractable LinHC: Only MLWE

The construction of our first extractable LinHC protocol based on MLWE is provided in Figure 4.

$\underline{\mathsf{KeyGen}^{\mathsf{H}}(1^\kappa)}$

1: $\rho \leftarrow \{0,1\}^\nu$
2: $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{H}(\rho)$
3: **return** $\mathsf{K}_{\mathsf{com}} := (\mathbf{A}, \mathbf{B}) \in R_q^{m\times n} \times R_q^{m\times n}$

$\underline{\mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})}$

1: **for** $i \in [N]$ **do**
2: $\quad (\mathbf{s}_{i,1}, \mathbf{s}_{i,2}, \mathbf{s}_{i,3}) \leftarrow S_\eta^n \times S_\eta^m \times S_\eta^m$
3: $\quad \mathbf{t}_{i,1} \leftarrow p \cdot (\mathbf{A}\mathbf{s}_{i,1} + \mathbf{s}_{i,2})$
4: $\quad \mathbf{t}_{i,2} \leftarrow p \cdot (\mathbf{B}\mathbf{s}_{i,1} + \mathbf{s}_{i,3}) + \mathbf{e}_i$
5: $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \leftarrow D_{\phi\cdot T}^n \times D_{\phi\cdot T}^m \times D_{\phi\cdot T}^m$
6: $\mathbf{w}_1 \leftarrow p \cdot (\mathbf{A}\mathbf{y}_1 + \mathbf{y}_2)$
7: $\mathbf{w}_2 \leftarrow p \cdot (\mathbf{B}\mathbf{y}_1 + \mathbf{y}_3) + \mathbf{r}$
8: $\mathsf{com} := \big((\mathbf{t}_{i,1}, \mathbf{t}_{i,2})_{i\in[N]}, \mathbf{w}_1, \mathbf{w}_2\big)$
9: $\mathsf{st} := \big((\mathbf{s}_{i,1}.\mathbf{s}_{i,2}, \mathbf{s}_{i,3})_{i\in[N]}, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\big)$
10: **return** $(\mathsf{com}, \mathsf{st})$

$\underline{\mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \mathbf{z}), \mathsf{st}))}$

1: $(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
2: **for** $\ell \in \{1,2,3\}$ **do**
3: $\quad \bar{\mathbf{s}}_\ell \leftarrow \sum_{i=1}^N \beta_i \cdot \mathbf{s}_{i,\ell}$
4: $\quad \mathbf{z}_\ell \leftarrow \bar{\mathbf{s}}_\ell + \mathbf{y}_\ell$
5: $b \leftarrow \mathsf{Rej}([\mathbf{z}_1\|\mathbf{z}_2\|\mathbf{z}_3], [\bar{\mathbf{s}}_1\|\bar{\mathbf{s}}_2\|\bar{\mathbf{s}}_3], \phi, T, \mathsf{err})$
6: **if** $b = \perp$ **then return** $\mathsf{op} := \perp$
7: **else return** $\mathsf{op} := [\mathbf{z}_1\|\mathbf{z}_2\|\mathbf{z}_3]$

$\underline{\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op} \neq \perp)))}$

1: $(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
2: $\big(\mathbf{z}_{\mathbf{r}}, (\mathbf{t}_{i,1}.\mathbf{t}_{i,2})_{i\in[N]}, \mathbf{w}_1, \mathbf{w}_2\big) \leftarrow \mathsf{com}$
3: $[\mathbf{z}_1\|\mathbf{z}_2\|\mathbf{z}_3] \leftarrow \mathsf{op}$
4: **for** $\ell \in \{1,2,3\}$ **do**
5: $\quad$ **if** $\|\mathbf{z}_\ell\|_2 > \sqrt{2nd} \cdot \phi \cdot T$ **then return** $\perp$
6: $\mathbf{z}_{\mathbf{A}} \leftarrow \sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1 - p \cdot (\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2)$
7: $\mathbf{z}_{\mathbf{B}} \leftarrow \sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2 - p \cdot (\mathbf{B}\mathbf{z}_1 + \mathbf{z}_3)$
8: **if** $\mathbf{z}_{\mathbf{A}} \neq \mathbf{0} \ \vee \ \mathbf{z} \neq \mathbf{z}_{\mathbf{B}}$ **then return** $\perp$
9: **else return** $\top$

Figure 4: An extractable LinHC protocol based on MLWE.

Parameters and size. Let the dimension d of the ring R_q be larger than 256 and n, m be positive integers such that $n \leq m$,[18] $p < q$ be coprime odd integers, η a positive real, and H be a random oracle with domain $\{0,1\}^\nu$ and range $R_q^{m\times n} \times R_q^{m\times n}$. The concrete value of ν is specific to the underlying Σ-protocol being used. Let T, ϕ, and err be parameters required by the rejection sampling algorithm Lemma 2.10, where we set $T = \eta \cdot \sum_{i=1}^N \|\beta_i\|_\infty \cdot \sqrt{(n+2m)d}$.

The size of the first message com is $2md(N+1)\log q$ and the third message op is $(n+2m)d \cdot \log(10\phi T)$. Looking ahead, when we make the protocol non-interactive via the Fiat-Shamir transform, we can send the challenge $\boldsymbol{\beta}$ instead of $(\mathbf{w}_1, \mathbf{w}_2)$ since the latter can be recovered from the other components and $\boldsymbol{\beta}$. In this case, the total size is $2mdN\log q + (n+2m)d \cdot \log(10\phi T) + |\mathsf{ChSet}|$.

Properties. The following Lemmata 3.11 to 3.13 establishes the correctness and security of our extractable LinHC. In Section 3.6, we discuss the simplified version of our extractable LinHC and see that we only require MLWE instead of the quantum accessible MLWE.

Lemma 3.11 (Correctness). *The extractable* LinHC *protocol in Figure 4 has correctness error* (δ_0, δ_1) *with* $\delta_0 \leq 2^{-256}$ *and* $\delta_1 = 1 - (1 - \mathsf{err})/\mu(\phi, \mathsf{err})$. *For instance, if* $\phi = 14$ *and* $\mathsf{err} = 2^{-256}$, *then* $\delta_1 \approx 3/4$.

Proof. First, the probability of $\mathsf{op} = \perp$ for an honest execution of the protocol is δ_1 due to Lemma 2.10. We now check the correctness of the verification algorithm conditioned on $\mathsf{op} \neq \perp$. By setting $t = \sqrt{2}$ in Lemma 2.11 and using the assumption that $d \geq 256$, the check in Line 5 of the verification algorithm will pass with probability at least $1 - \delta_0$. Moreover, routine calculation shows that Line 8 of the verification algorithm also holds. That is, $\mathbf{z}_{\mathbf{A}} = \mathbf{0}$ and $\mathbf{z} = \sum_{i=1}^N \beta_i \cdot \mathbf{e}_i + \mathbf{r} = \mathbf{z}_{\mathbf{B}}$. Therefore, correctness holds. □

[18] d could be set arbitrary as long as the underling hardness assumptions (MLWE and DSMR) hold. We consider a lower bound of 256 to make it easier to provide concrete bounds on the properties of extractable LinHC, e.g., Lemma 3.11.

Lemma 3.12 (QAnaHVZK). *Define the zero-knowledge simulator* ZKSim *as in Figure 5. Then, for any QPT adversary* $\mathcal{A}$ *against* $\mathsf{QAnaHVZK}$ *of the extractable* LinHC *protocol in Figure 4 making at most* Q *oracle queries, there exists a QPT adversary* $\mathcal{B}$ *against the quantum accessible* $\mathsf{MLWE}_{n,m,2^\nu,Q,\eta}$ *problem such that*

$$\mathsf{Adv}^{\mathsf{QAnaHVZK}}(\mathcal{A}) \leq N \cdot \mathsf{Adv}^{\mathsf{qaMLWE}_{n,2m,2^\nu,Q,\eta}}(\mathcal{B}) + \sqrt{C \cdot Q^3 \cdot \frac{\mathsf{err}}{\mu(\phi,\mathsf{err})}},$$

where $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$. *Here* C *is a positive constant defined independent of* $\mathcal{A}$.

Proof. Fix an arbitrary $(\mathbf{e}_i)_{i\in[N]} \in (R_q^m)^N$. We assume for simplicity that when $\mathcal{A}$ makes a quantum query $\sum_\rho \alpha_\rho |\rho\rangle$ to its oracles, it receives both $\sum_\rho \alpha_\rho |\mathsf{H}(\rho)\rangle$ and $\sum_\rho \alpha_\rho |D(\rho,(\mathbf{e}_i)_{i\in[N]})\rangle$. Now define a simulator ZKSim_0 that on input $\mathsf{K}_{\mathsf{com}}, \boldsymbol{\beta} \in \mathsf{ChSet}$, and $\mathsf{W} := ((\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$ outputs a transcript identical to an honest execution conditioned on $\mathsf{op} \neq \bot$ and the verifier outputting $\boldsymbol{\beta}$. Then, $D^{\not\downarrow}_{\mathsf{trans}}(\rho,(\mathbf{e}_i)_{i\in[N]})$ is equivalent to the distribution that first generates $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}}(1^\kappa)[\rho]$, samples $(\boldsymbol{\beta},\mathbf{r}) \leftarrow D_{\boldsymbol{\beta},\mathbf{r}}$, runs $(\mathsf{com},\mathsf{op}) \leftarrow \mathsf{ZKSim}_0(\mathsf{K}_{\mathsf{com}},\boldsymbol{\beta},\mathsf{W})$, and finally outputs $(\mathbf{r},\mathsf{trans}=(\mathsf{com},\boldsymbol{\beta},(\mathbf{z},\mathsf{op})))$. On the other hand, D_{sim} is the same as $D^{\not\downarrow}_{\mathsf{trans}}$ except that it runs ZKSim instead of ZKSim_0, where ZKSim only takes $(\mathsf{K}_{\mathsf{com}},\boldsymbol{\beta},\mathbf{z})$ as input. Therefore, it suffices to show that ZKSim_0 and ZKSim are indistinguishable even given oracle access to them. In the following, let us consider a sequence of simulators ZKSim_i, where $\mathsf{ZKSim}_2 := \mathsf{ZKSim}$. We show that each adjacent simulators are indistinguishable.

$\mathsf{ZKSim}_0(\mathsf{K}_{\mathsf{com}},\boldsymbol{\beta},\mathsf{W})$: It computes $\mathbf{z} = \sum_{i=1}^N \beta_i \cdot \mathbf{e}_i + \mathbf{r}$, runs $(\mathsf{com},\mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}},\mathsf{W})$, $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}},(\mathsf{com},\boldsymbol{\beta},\mathbf{z}),\mathsf{st})$, and finally outputs $(\mathsf{com},\mathsf{op})$ conditioned on $\mathsf{op} \neq \bot$.

$\mathsf{ZKSim}_1(\mathsf{K}_{\mathsf{com}},\boldsymbol{\beta},\mathsf{W})$: We modify how it computes $\mathsf{op} = (\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3)$. Specifically, it samples $(\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3) \leftarrow D^n_{\phi\cdot T} \times D^m_{\phi\cdot T} \times D^m_{\phi\cdot T}$, and sets

$$\mathbf{w}_1 = -\sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,1} + p \cdot (\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2), \quad \mathbf{w}_2 = -\sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,2} + p \cdot (\mathbf{B}\mathbf{z}_1 + \mathbf{z}_3) + \mathbf{z},$$

where $(\mathbf{t}_{i,1},\mathbf{t}_{i,2})_{i\in[N]}$ are set as in ZKSim_0. It then outputs $(\mathsf{com},\mathsf{op})$.

Due to Lemma 2.10, conditioned on $\mathsf{op} \neq \bot$, the distribution of the transcripts output by ZKSim_0 and ZKSim_1 are within statistical distance $\mathsf{err}/\mu(\phi,\mathsf{err})$. Moreover, notice ZKSim_0 and ZKSim_1 are respectively distributed independently and identically for each input (which is randomly defined by $\rho \in \{0,1\}^\nu$). Therefore, by Lemma 2.8, the advantage of distinguishing ZKSim_0 and ZKSim_1 even with oracle access to them is bounded by $\sqrt{C \cdot Q^3 \cdot \mathsf{err}/\mu(\phi,\mathsf{err})}$ for some universal constant $C > 0$. The transcript is now independent of $\mathbf{r}$.

$\mathsf{ZKSim}_2(\mathsf{K}_{\mathsf{com}},\boldsymbol{\beta},\mathsf{W})$: We modify how it computes $(\mathbf{t}_{i,1},\mathbf{t}_{i,2})_{i\in[N]}$. Specifically, it samples $(\mathbf{t}_{i,1},\mathbf{t}_{i,2}) \leftarrow R_q^m \times R_q^m$ for all $i \in [N]$. All other terms are set as in ZKSim_1. It then outputs $(\mathsf{com},\mathsf{op})$. Notice that ZKSim_2 is identical to ZKSim.

Indistinguishability of the transcripts output by ZKSim_1 and ZKSim_2 follows from a simple hybrid argument using the quantum accessible $\mathsf{MLWE}_{n,2m,2^\nu,Q,\eta}$ assumption. This follows by noticing that due to the modification we made in ZKSim_1, $(\mathbf{s}_{i,1},\mathbf{s}_{i,2},\mathbf{s}_{i,3})_{i\in[N]}$ are now independent of $(\mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3)$. The qaMLWE adversary $\mathcal{B}$ will be able to simulate the transcript by embedding its challenge into $(\mathbf{t}_{i,1},\mathbf{t}_{i,2})_{i\in[N]}$. In case the MLWE sample is valid (resp. random), it simulates ZKSim_1 (resp. ZKSim_2) perfectly. Since there are N-tuples $(\mathbf{t}_{i,1},\mathbf{t}_{i,2})$, we can go through N hybrid arguments. Moreover, it is clear that the running time of $\mathbf{B}$ is closely related to $\mathbf{A}$. Finally, notice that we can simulate the random oracle H by using the oracle provided by the MLWE problem.

This completes the proof. □

$\underline{\mathsf{ZKSim}(\mathsf{K}_{\mathsf{com}}, \boldsymbol{\beta}, \mathbf{z})}$

$(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) \leftarrow D^n_{\phi \cdot T} \times D^m_{\phi \cdot T} \times D^m_{\phi \cdot T}$
for $i \in [N]$ **do**
  $(\mathbf{t}_{i,1}, \mathbf{t}_{i,2}) \leftarrow R^m_q \times R^m_q$
$\mathbf{w}_1 \leftarrow -\sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,1} + p \cdot (\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2)$
$\mathbf{w}_2 \leftarrow -\sum_{i=1}^N \beta_i \cdot \mathbf{t}_{i,2} + p \cdot (\mathbf{B}\mathbf{z}_1 + \mathbf{z}_3) + \mathbf{z}$
$\mathsf{com} := \big(\mathbf{z}_{\mathbf{r}}, (\mathbf{t}_{i,1}, \mathbf{t}_{i,2})_{i \in [N]}, \mathbf{w}_1, \mathbf{w}_2\big)$
$\mathsf{op} := [\mathbf{z}_1 \| \mathbf{z}_2 \| \mathbf{z}_3]$
return $(\mathsf{com}, \mathsf{op})$

$\underline{\widetilde{\mathsf{H}}(\rho)}$

$(\rho_1, \rho_2, \rho_3) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
$\mathbf{A} \leftarrow R^{m \times n}_q[\rho_1]$
$(\mathbf{D}_1, \mathbf{D}_2) \leftarrow S^{m \times m}_\eta[\rho_2] \times S^{m \times n}_\eta[\rho_3]$
$\mathbf{B} \leftarrow \mathbf{D}_1\mathbf{A} + \mathbf{D}_2$
return $(\mathbf{A}, \mathbf{B})$

$\underline{\mathsf{SimOracle}(1^\kappa)}$

$\mathsf{K} \leftarrow \mathcal{K}$ ▷ Sample PRF key
return $(\widetilde{\mathsf{H}}, \tau := \mathsf{K})$

$\underline{\mathsf{LinCExtract}(\tau = \mathsf{K}, \rho, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))}$

$(\rho_1, \rho_2, \rho_3) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
$\mathbf{D}_1 \leftarrow S^{m \times m}_\eta[\rho_2]$
$\big((\mathbf{t}_{i,1}.\mathbf{t}_{i,2})_{i \in [N]}, \mathbf{w}_1, \mathbf{w}_2\big) \leftarrow \mathsf{com}$
$(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
$(c, L) \leftarrow (0, \{(\boldsymbol{\beta}, \mathbf{z})\})$
while $|L| \le k \ \vee \ c \le T^*$ **do**
  $\widetilde{\boldsymbol{\beta}} = (\widetilde{\beta}_1, \cdots, \widetilde{\beta}_N) \leftarrow \mathsf{ChSet} \backslash L_{\boldsymbol{\beta}}$
  $\widetilde{\mathbf{z}} \leftarrow \big(\sum_{i=1}^N \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2\big)$
    $-\mathbf{D}_1\big(\sum_{i=1}^N \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1\big) \mod p$
  if $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ **then** $L \leftarrow L \cup \{(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}})\}$
  $c \leftarrow c + 1$
if $|L| < k$ **then return** $\perp$
else return L

Figure 5: Description of ZKSim, SimOracle, $\widetilde{\mathsf{H}}$, and LinCExtract for the extractable LinHC protocol in Figure 4. Here the PRF key K is assumed to be hardwired to $\widetilde{\mathsf{H}}$ and denote $L_{\boldsymbol{\beta}}$ as the set $\{\boldsymbol{\beta} \mid (\boldsymbol{\beta}, \mathbf{z}) \in L\}$.

Lemma 3.13 ($\mathcal{F}_B$-Almost straight-line extractable). *Assume* $B \ge \sqrt{2nd} \cdot \phi \cdot T$, $2\sqrt{2}p(nd\eta + \sqrt{nmd}\eta + \sqrt{nd})\phi T + 2B < q/2$, *and* $B \le (p-1)/4$. *Define the oracle simulator* SimOracle *and linear commitment extractor* LinCExtract *as in Figure 5, where* T^* *in Line 6 of algorithm* LinCExtract *is* $T^* = \frac{k \cdot \delta \cdot |\mathsf{ChSet}|}{|S_f(\rho, \mathsf{com})| - k}$. *Then, the extractable* LinHC *protocol in Figure 4 is* $\mathcal{F}_B$*-almost straight-line extractable also satisfying the optional properties in Definition 3.5. Moreover, for any QPT adversary* $\mathcal{A}$ *that distinguishes between a random* H *and* $\widetilde{\mathsf{H}}$ *output by* SimOracle *making at most* Q *queries, there exists a QPT adversary* $\mathcal{B}_1$ *against the quantum accessible* $\mathsf{MLWE}_{m,n,2^\nu,Q,\eta}$ *problem and a QPT adversary* $\mathcal{B}_2$ *against the quantum accessible* PRF *such that*

$$\mathsf{Adv}^{\mathsf{IndO}}(\mathcal{A}) \le m \cdot \mathsf{Adv}^{\mathsf{qaMLWE}_{m,n,2^\nu,Q,\eta}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{qaPRF}}(\mathcal{B}_2),$$

where $\mathsf{Time}(\mathcal{A}) = \mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{B}_2)$.

Proof. We prove Items 1 and 2 in Definition 3.4 and the two optional Items 3 and 4 in Definition 3.5.

Item 1. Consider the following sequence of oracles H_i, where H_0 is identical to H and H_2 is identical to $\widetilde{\mathsf{H}}$. We show that each adjacent oracles are indistinguishable.

- H_0 : Same as H. That is, a random function is sampled.
- H_1 : We modify H_0 so that it outputs a random MLWE sample. Concretely, sample a random function $\mathsf{G} : \{0,1\}^\nu \to R^{m \times n}_q \times S^{m \times m}_\eta \times S^{m \times n}_\eta$ and define $\mathsf{H}_1(\rho)$ to run $\mathsf{G}(\rho) \to (\mathbf{A}, \mathbf{D}_1, \mathbf{D}_2)$ and then to output $(\mathbf{A}, \mathbf{B} = \mathbf{D}_1\mathbf{A} + \mathbf{D}_2)$. By invoking the quantum accessible $\mathsf{MLWE}_{m,n,2^\nu,Q,\eta}$ assumption for each row of $\mathbf{B}$ we conclude that H_0 and H_1 are indistinguishable.
- H_2 : We modify H_1 to use a $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\nu \to R^{m \times n}_q \times S^{m \times m}_\eta \times S^{m \times n}_\eta$ instead of G to sample $(\mathbf{A}, \mathbf{D}_1, \mathbf{D}_2)$. Due to the security of the quantum accessible PRF, H_1 and H_2 are indistinguishable.

This completes the proof of Item 1.

Item 2. Fix any $(\widetilde{\mathsf{H}}, \tau = \mathsf{K})$, randomness $\rho \in \{0,1\}^\nu$, first message $\mathsf{com} = \big((\mathbf{t}_{i,1}, \mathbf{t}_{i,2})_{i \in [N]}, \mathbf{w}_1, \mathbf{w}_2\big)$, and any function $f \in \mathcal{F}_B$. Moreover, let $\mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ be a valid transcript. We first show that

conditioned on $\widetilde{\boldsymbol{\beta}} \in S_f(\rho, \mathsf{com}) \backslash \{\boldsymbol{\beta}\} \subset \mathsf{ChSet}$ being sampled in Line 7, $\mathsf{LinCExtract}(\tau, \rho, \mathsf{trans})$ always succeeds in outputting a valid $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$. By definition of the set $S_f(\rho, \mathsf{com})$, existence of $(\widetilde{\mathbf{z}}, \widetilde{\mathsf{op}})$ such that $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \widetilde{\boldsymbol{\beta}}, (\widetilde{\mathbf{z}}, \widetilde{\mathsf{op}}))) = \top$ and $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ is guaranteed. Therefore, denoting $\widetilde{\mathsf{op}} = [\widetilde{\mathbf{z}}_1 \| \widetilde{\mathbf{z}}_2 \| \widetilde{\mathbf{z}}_3]$, we have $\|\widetilde{\mathbf{z}}_\ell\|_2 \leq \sqrt{2nd} \cdot \phi \cdot T$ for all $\ell \in \{1, 2, 3\}$, and

$$p \cdot (\mathbf{A}\widetilde{\mathbf{z}}_1 + \widetilde{\mathbf{z}}_2) = \sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1, \quad p \cdot (\mathbf{B}\widetilde{\mathbf{z}}_1 + \widetilde{\mathbf{z}}_3) + \widetilde{\mathbf{z}} = \sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2,$$

where $\mathbf{A}$ and $\mathbf{B} = \mathbf{D}_1\mathbf{A} + \mathbf{D}_2$ are uniquely defined by $\widetilde{\mathsf{H}}(\rho)$ and $\tau = \mathsf{K}$ as in Figure 5. Therefore,

$$\begin{aligned}
&\left\| \Big(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2 \Big) - \mathbf{D}_1 \Big(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1 \Big) \right\|_\infty \\
=& \left\| p \cdot \Big(\mathbf{D}_2\widetilde{\mathbf{z}}_1 - \mathbf{D}_1\widetilde{\mathbf{z}}_2 + \widetilde{\mathbf{z}}_3 \Big) + \widetilde{\mathbf{z}} \right\|_\infty \\
\leq& p \cdot \left(\sqrt{nd} \|\mathbf{D}_2\|_\infty \cdot \|\widetilde{\mathbf{z}}_1\|_2 + \sqrt{md} \|\mathbf{D}_1\|_\infty \cdot \|\widetilde{\mathbf{z}}_2\|_2 + \|\widetilde{\mathbf{z}}_3\|_\infty \right) + \|\widetilde{\mathbf{z}}\|_\infty \\
\leq& \sqrt{2}p(nd\eta + \sqrt{nmd}\eta + \sqrt{nd})\phi T + B < q/2,
\end{aligned}$$

where we have $\|\widetilde{\mathbf{z}}\|_2 \leq B$ by definition of $\mathcal{F}_B$ (see Section 3.3), $\|\mathbf{D}_1\|_\infty, \|\mathbf{D}_2\|_\infty \leq \eta$, and the last equation holds from the assumption in the statement. Moreover, we use the fact that for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$, we have $\|\mathbf{a}^\top \mathbf{b}\|_\infty \leq \sqrt{n}\|\mathbf{a}\|_\infty \|\mathbf{b}\|_2$. This implies that the equality holds over R, and in particular, when $\|\widetilde{\mathbf{z}}\|_\infty \leq B \leq (p-1)/2$, $\big(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2 \big) - \mathbf{D}_1 \big(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1 \big) \mod p$ is identical to $\widetilde{\mathbf{z}}$. Hence, we are able to extract $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$.

Next, we check that $\mathsf{LinCExtract}$ succeeds in outputting a set $L = \{(\widetilde{\boldsymbol{\beta}}_j, \widetilde{\mathbf{z}}_j)\}_{j \in [k]}$ such that $f(\widetilde{\boldsymbol{\beta}}_j, \widetilde{\mathbf{z}}_j) = \top$ for all $j \in [k]$, where by construction all the $\widetilde{\boldsymbol{\beta}}_j$'s are pairwise distinct. Since $\widetilde{\boldsymbol{\beta}}$ is sampled uniformly random from $\mathsf{ChSet} \backslash L_{\boldsymbol{\beta}}$, the probability of sampling $\widetilde{\boldsymbol{\beta}} \in S_f(\rho, \mathsf{com}) \backslash L_{\boldsymbol{\beta}}$ in one loop is at least $\frac{|S_f(\rho, \mathsf{com})| - k}{|\mathsf{ChSet}|}$. Therefore, given any L, if we sample $\widetilde{\boldsymbol{\beta}}$ $\frac{\delta \cdot |\mathsf{ChSet}|}{|S_f(\rho, \mathsf{com})| - k}$-times from the set $\mathsf{ChSet} \backslash L_{\boldsymbol{\beta}}$, then the probability of sampling $\widetilde{\boldsymbol{\beta}} \in S_f(\rho, \mathsf{com}) \backslash L_{\boldsymbol{\beta}}$ is at least $1 - 2^{-\delta}$. Since each loop is independent from each other, after $T^* = \frac{k \cdot \delta \cdot |\mathsf{ChSet}|}{|S_f(\rho, \mathsf{com})| - k}$-loops, we obtain the desired set L with probability at least $1 - k \cdot 2^{-\delta}$, where the bound follows from the union bound. Finally, since each loop takes a fixed polynomial time, the running time of $\mathsf{LinCExtract}$ is $T^* \cdot \mathsf{poly}(\kappa)$ as desired. We note that there could exist $\widetilde{\boldsymbol{\beta}} \notin S_f(\rho, \mathsf{com})$ for which $\mathsf{LinCExtract}$ succeeds in extracting $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$. However, this will not be a problem since such $\widetilde{\boldsymbol{\beta}}$ can only increase the success probability and lower the running time of $\mathsf{LinCExtract}$.

This completes the proof of Item 2.

In case ChSet is only of polynomial size we define $\mathsf{LinCExtract}$ to run for all $\widetilde{\beta} \in \mathsf{ChSet}$ in Figure 5. Then, it is clear that Item 3 holds since we can enumerate over all $\widetilde{\beta} \in \mathsf{ChSet}$ in polynomial time and $\mathsf{LinCExtract}$ always output $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ when $\widetilde{\boldsymbol{\beta}} \in S_f(\rho, \mathsf{com})$.

Item 4. Fix any $(\widetilde{\mathsf{H}}, \tau = \mathsf{K})$, randomness $\rho \in \{0,1\}^\nu$, first message $\mathsf{com} = \big((\mathbf{t}_{i,1}, \mathbf{t}_{i,2})_{i \in [N]}, \mathbf{w}_1, \mathbf{w}_2 \big)$. Assume there exists $\boldsymbol{\beta} \in \mathsf{ChSet}$ and $(\mathbf{z}, \mathbf{z}', \mathsf{op}, \mathsf{op}')$ satisfying $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, \mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})) = \top$ and $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, \mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}', \mathsf{op}')) = \top$, where $\mathsf{K}_{\mathsf{com}} = \mathsf{KeyGen}^{\widetilde{\mathsf{H}}}(1^\kappa)[\rho]$. Denote $\mathsf{op} = [\mathbf{z}_1 \| \mathbf{z}_2 \| \mathbf{z}_3]$ and $\mathsf{op}' = [\mathbf{z}'_1 \| \mathbf{z}'_2 \| \mathbf{z}'_3]$. Then, we have

$$\begin{aligned}
p \cdot (\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2) &= \sum_{i=1}^{N} \beta_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1, \quad p \cdot (\mathbf{B}\mathbf{z}_1 + \mathbf{z}_3) + \mathbf{z} = \sum_{i=1}^{N} \beta_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2, \\
p \cdot (\mathbf{A}\mathbf{z}'_1 + \mathbf{z}'_2) &= \sum_{i=1}^{N} \beta_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1, \quad p \cdot (\mathbf{B}\mathbf{z}'_1 + \mathbf{z}'_3) + \mathbf{z}' = \sum_{i=1}^{N} \beta_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2.
\end{aligned}$$

By subtracting the two sides and substituting $\mathbf{B} = \mathbf{D}_1\mathbf{A} + \mathbf{D}_2$, we have

$$p \cdot \Big(\mathbf{D}_2(\mathbf{z}_1 - \mathbf{z}'_1) - \mathbf{D}_1(\mathbf{z}_2 - \mathbf{z}'_2) + (\mathbf{z}_3 - \mathbf{z}'_3)\Big) + (\mathbf{z} - \mathbf{z}') = \mathbf{0}$$

Following the same argument we made in the proof for Item 2, the left hand side holds over R (and not only over R_q). Therefore, taking mod p over both sides, we have $\mathbf{z} - \mathbf{z}' = \mathbf{0} \mod p$. Since $\|\mathbf{z} - \mathbf{z}'\|_\infty \leq 2B \leq (p-1)/2$, $\mathbf{z} - \mathbf{z}' = \mathbf{0}$ holds over R as well. This completes the proof of Item 4. □

Remark 3.14 (Using Grover's Algorithm). We can get an asymptotically more efficient extractor by allowing algorithm $\mathsf{LinCExtract}$ to perform quantum computation. At a high level, our classical algorithm $\mathsf{LinCExtract}$ is simply searching for an element in the set ChSet that satisfies an efficiently computable predicate $f(\cdot,\cdot)$. In the classical setting, the best we can do is to sample a random element in ChSet and hope that it satisfies this predicate. On the other hand, if we allow $\mathsf{LinCExtract}$ to be a quantum algorithm, we can get a quadratic speed up by using the Grover's search algorithm [Gro96]. In particular, the runtime of $\mathsf{LinCExtract}$ can be lowered down to roughly $k \cdot \sqrt{\frac{|\mathsf{ChSet}|}{|S_f(\rho,\mathsf{com})|-k}}$ from $\frac{k\cdot|\mathsf{ChSet}|}{|S_f(\rho,\mathsf{com})|-k}$.

3.5 Second Construction of Extractable LinHC: MLWE + DSMR

The second construction of our extractable LinHC protocol based on LWE and DSMR is provided in Figure 6. The high level structure of the protocol is the same as in our first construction. The only difference is how we "encrypt" the witness vectors $(\mathbf{e}_i)_{i\in[N]}$. Namely, by using an NTRU-type encryption, we are able to halve the proof size compared to our first construction.

$\underline{\mathsf{KeyGen}^{\mathsf{H}}(1^\kappa)}$
1: $\rho \leftarrow \{0,1\}^\nu$
2: $\mathbf{H} \leftarrow \mathsf{H}(\rho)$
3: **return** $\mathsf{K}_{\mathsf{com}} := \mathbf{H} \in R_q^{m\times n}$

$\underline{\mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})}$
1: **for** $i \in [N]$ **do**
2: $\quad(\mathbf{s}_{i,1}, \mathbf{s}_{i,2}) \leftarrow S_\eta^n \times S_\eta^m$
3: $\quad\mathbf{t}_i \leftarrow \mathbf{H}\mathbf{s}_{i,1} + p\cdot\mathbf{s}_{i,2} + \mathbf{e}_i$
4: $(\mathbf{y}_1, \mathbf{y}_2) \leftarrow D_{\phi\cdot T}^n \times D_{\phi\cdot T}^m$
5: $\mathbf{w} \leftarrow \mathbf{H}\mathbf{y}_1 + p\cdot\mathbf{y}_2 + \mathbf{r}$
6: $\mathsf{com} := \big((\mathbf{t}_i)_{i\in[N]}, \mathbf{w}\big)$
7: $\mathsf{st} := \big((\mathbf{s}_{i,1}.\mathbf{s}_{i,2})_{i\in[N]}, \mathbf{y}_1, \mathbf{y}_2\big)$
8: **return** $(\mathsf{com}, \mathsf{st})$

$\underline{\mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \mathbf{z}), \mathsf{st}))}$
1: $(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
2: **for** $\ell \in \{1,2\}$ **do**
3: $\quad\bar{\mathbf{s}}_\ell \leftarrow \sum_{i=1}^N \beta_i \cdot \mathbf{s}_{i,\ell}$
4: $\quad\mathbf{z}_\ell \leftarrow \bar{\mathbf{s}}_\ell + \mathbf{y}_\ell$
5: $b \leftarrow \mathsf{Rej}([\mathbf{z}_1\|\mathbf{z}_2], [\bar{\mathbf{s}}_1\|\bar{\mathbf{s}}_2], \phi, T, \mathsf{err})$
6: **if** $b = \perp$ **then return** $\mathsf{op} := \perp$
7: **else return** $\mathsf{op} := [\mathbf{z}_1\|\mathbf{z}_2]$

$\underline{\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op} \neq \perp)))}$
1: $(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
2: $\big((\mathbf{t}_i)_{i\in[N]}, \mathbf{w}\big) \leftarrow \mathsf{com}$
3: $[\mathbf{z}_1\|\mathbf{z}_2] \leftarrow \mathsf{op}$
4: **for** $\ell \in \{1,2\}$ **do**
5: $\quad$**if** $\|\mathbf{z}_\ell\|_2 > \sqrt{2nd}\cdot\phi\cdot T$ **then return** $\perp$
6: $\mathbf{z}_{\mathbf{H}} \leftarrow \sum_{i=1}^N \beta_i \cdot \mathbf{t}_i + \mathbf{w} - (\mathbf{H}\mathbf{z}_1 + p\cdot\mathbf{z}_2)$
7: **if** $\mathbf{z} \neq \mathbf{z}_{\mathbf{H}}$ **then return** $\perp$
8: **else return** $\top$

Figure 6: An extractable LinHC protocol based on MLWE and DSMR.

Parameters and size. The parameters are identical to those discussed in our first construction. The size of the first message com is $md(N+1)\log q$ and the third message op is $(n+m)d\cdot\log(10\phi T)$. When we make the protocol non-interactive via the Fiat-Shamir transform, we can send the challenge $\boldsymbol{\beta}$ instead of $\mathbf{w}$ since the latter can be recovered from the other components and $\boldsymbol{\beta}$. In this case, the total size is $mdN\log q + (n+m)d\cdot\log(10\phi T) + |\mathsf{ChSet}|$. Since the $\mathbf{t}$'s in com are the dominant term, the second construction is roughly half the size of our first construction.

Properties. The following Lemmata 3.15 to 3.17 establishes the correctness and security of our extractable LinHC protocol. We omit the proof of Lemmata 3.15 and 3.16 since they are identical to those of the first construction.

Lemma 3.15 (Correctness). *The extractable* LinHC *protocol in Figure 6 has correctness error* (δ_0, δ_1) *with* $\delta_0 \leq 2^{-256}$ *and* $\delta_1 = 1 - (1 - \mathsf{err})/\mu(\phi, \mathsf{err})$. *For instance, if* $\phi = 14$ *and* $\mathsf{err} = 2^{-256}$, *then* $\delta_1 \approx 3/4$.

Lemma 3.16 (QAnaHVZK). *Define the zero-knowledge simulator* ZKSim *as in Figure 7. Then, for any QPT adversary* $\mathcal{A}$ *against* QAnaHVZK *of the extractable* LinHC *protocol in Figure 6 making at most* Q *oracle queries, there exists a QPT adversary* $\mathcal{B}$ *against the quantum accessible* $\mathsf{MLWE}_{n,m,2^\nu,Q,\eta}$ *problem such that*

$$\mathsf{Adv}^{\mathsf{QAnaHVZK}}(\mathcal{A}) \leq N \cdot \mathsf{Adv}^{\mathsf{qaMLWE}_{n,m,2^\nu,Q,\eta}}(\mathcal{B}) + \sqrt{C \cdot Q^3 \cdot \frac{\mathsf{err}}{\mu(\phi, \mathsf{err})}},$$

where $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$. *Here* C *is a positive constant defined independent of* $\mathcal{A}$.

$\underline{\mathsf{ZKSim}(\mathsf{K}_{\mathsf{com}}, \boldsymbol{\beta}, \mathbf{z})}$

$(\mathbf{z}_1, \mathbf{z}_2) \leftarrow D^n_{\phi \cdot T} \times D^m_{\phi \cdot T}$
for $i \in [N]$ **do**
  $\mathbf{t}_i \leftarrow R^m_q$
$\mathbf{w} \leftarrow -\sum_{i=1}^N \beta_i \cdot \mathbf{t}_i + (\mathbf{H}\mathbf{z}_1 + p \cdot \mathbf{z}_2)$
$\mathsf{com} := \big((\mathbf{t}_i)_{i \in [N]}, \mathbf{w}\big)$
$\mathsf{op} := [\mathbf{z}_1 \| \mathbf{z}_2]$
return $(\mathsf{com}, \mathsf{op})$

$\underline{\widetilde{\mathsf{H}}(\rho)}$

$(\rho_1, \rho_2) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
$(\mathbf{F}, \mathbf{G}) \leftarrow \hat{S}^{m \times m}_\eta[\rho_1] \times S^{m \times n}_\eta[\rho_2]$
$\mathbf{H} \leftarrow p \cdot \mathbf{F}^{-1}\mathbf{G}$
return $\mathbf{H}$

$\underline{\mathsf{SimOracle}(1^\kappa)}$

$\mathsf{K} \leftarrow \mathcal{K}$ ▷ Sample PRF key
return $(\widetilde{\mathsf{H}}, \tau := \mathsf{K})$

$\underline{\mathsf{LinCExtract}(\tau = \mathsf{K}, \rho, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))}$

$(\rho_1, \rho_2) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
$\mathbf{F} \leftarrow \hat{S}^{m \times m}_\eta[\rho_2]$
$\big((\mathbf{t}_i)_{i \in [N]}, \mathbf{w}\big) \leftarrow \mathsf{com}$
$(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
$(c, L) \leftarrow (0, \{(\boldsymbol{\beta}, \mathbf{z})\})$
while $|L| \leq k \ \vee \ c \leq T^*$ **do**
  $\widetilde{\boldsymbol{\beta}} = (\widetilde{\beta}_1, \cdots, \widetilde{\beta}_N) \leftarrow \mathsf{ChSet} \backslash L_{\boldsymbol{\beta}}$
  $\widetilde{\mathbf{v}} \leftarrow \mathbf{F}\Big(\sum_{i=1}^N \widetilde{\beta}_i \cdot \mathbf{t}_i + \mathbf{w}\Big) \mod p$
  $\widetilde{\mathbf{z}} \leftarrow \mathbf{F}^{-1}\widetilde{\mathbf{v}} \mod p$
  if $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ **then** $L \leftarrow L \cup \{(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}})\}$
  $c \leftarrow c + 1$
if $|L| < k$ **then return** $\perp$
else return L

Figure 7: Description of ZKSim, SimOracle, $\widetilde{\mathsf{H}}$, and LinCExtract for the extractable LinHC protocol in Figure 6. Here the PRF key K is assumed to be hardwired to $\widetilde{\mathsf{H}}$, denote $\hat{S}^{m \times m}_\eta \subset S^{m \times m}_\eta$ be the set of all invertible elements over mod q and mod p, and denote $L_{\boldsymbol{\beta}}$ as the set $\{\boldsymbol{\beta} \mid (\boldsymbol{\beta}, \mathbf{z}) \in L\}$.

Lemma 3.17 ($\mathcal{F}_B$-Almost straight-line extractable). *Assume* $B \geq \sqrt{2nd} \cdot \phi \cdot T$, $2\sqrt{2}p(n + \sqrt{nm})d\eta\phi T + 2\sqrt{md}\eta B < q/2$, *and* $B \leq (p-1)/4$. *Define the oracle simulator* SimOracle *and linear commitment extractor* LinCExtract *as in Figure 7, where* T^* *in Line 6 of algorithm* LinCExtract *is* $T^* = \frac{k \cdot \delta \cdot |\mathsf{ChSet}|}{|S_f(\rho, \mathsf{com})| - k}$. *Then, the extractable* LinHC *protocol in Figure 6 is* $\mathcal{F}_B$*-almost straight-line extractable also satisfying the optional properties in Definition 3.5. Moreover, for any QPT adversary* $\mathcal{A}$ *that distinguishes between a random* H *and* $\widetilde{\mathsf{H}}$ *output by* SimOracle *making at most* Q *oracle queries, there exists a QPT adversary* $\mathcal{B}_1$ *against the quantum accessible* $\mathsf{DSMR}_{n,m,2^\nu,Q,\eta}$ *problem and a QPT adversary* $\mathcal{B}_2$ *against the quantum accessible* PRF *such that*

$$\mathsf{Adv}^{\mathsf{IndO}}(\mathcal{A}) \leq m \cdot \mathsf{Adv}^{\mathsf{qaDSMR}_{n,m,2^\nu,Q,\eta}}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{qaPRF}}(\mathcal{B}_2),$$

where $\mathsf{Time}(\mathcal{A}) = \mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{B}_2)$.

Proof. The proofs for Item 1 in Definition 3.4 is identical to the proof of our first construction in Lemma 3.13 except that we use the quantum accessible DSMR assumption instead of the quantum accessible MLWE assumption. Moreover, two optional Items 3 and 4 in Definition 3.5 follows naturally from Item 2 in Definition 3.4. Therefore, we only check Item 2 below.

Item 2. Fix any $(\widetilde{\mathsf{H}}, \tau = \mathsf{K})$, randomness $\rho \in \{0,1\}^\nu$, first message $\mathsf{com} = \big((\mathbf{t}_i)_{i\in[N]}, \mathbf{w}\big)$, and any function $f \in \mathcal{F}_B$. Moreover, let $\mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ be a valid transcript. Since the number of repetition required is identical to the first construction, we only need to show that conditioned on $\widetilde{\boldsymbol{\beta}} \in S_f(\rho, \mathsf{com})\backslash\{\boldsymbol{\beta}\} \subset \mathsf{ChSet}$ being sampled in Line 7, $\mathsf{LinCExtract}(\tau, \rho, \mathsf{trans})$ always succeeds in outputting a valid $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$. By definition of the set $S_f(\rho, \mathsf{com})$, existence of $(\widetilde{\mathbf{z}}, \widetilde{\mathsf{op}})$ such that $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \widetilde{\boldsymbol{\beta}}, (\widetilde{\mathbf{z}}, \widetilde{\mathsf{op}}))) = \top$ and $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ is guaranteed. Therefore, denoting $\widetilde{\mathsf{op}} = [\widetilde{\mathbf{z}}_1 \| \widetilde{\mathbf{z}}_2]$, we have $\|\widetilde{\mathbf{z}}_\ell\|_2 \le \sqrt{2nd} \cdot \phi \cdot T$ for all $\ell \in \{1,2\}$, and

$$\mathbf{H}\widetilde{\mathbf{z}}_1 + p \cdot \widetilde{\mathbf{z}}_2 + \widetilde{\mathbf{z}} = \sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_i + \mathbf{w},$$

where $\mathbf{H} = p \cdot \mathbf{F}^{-1}\mathbf{G}$ is uniquely defined by $\widetilde{\mathsf{H}}(\rho)$ and $\tau = \mathsf{K}$ as in Figure 7. Multiplying by $\mathbf{F}$ on the right hand side, we obtain

$$\begin{aligned}
\left\| \mathbf{F} \cdot \Big(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_i + \mathbf{w} \Big) \right\|_\infty &= \left\| p \cdot \Big(\mathbf{G}\widetilde{\mathbf{z}}_1 + \mathbf{F}\widetilde{\mathbf{z}}_2 \Big) + \mathbf{F}\widetilde{\mathbf{z}} \right\|_\infty \\
&\le p \cdot \big(\sqrt{nd} \|\mathbf{G}\|_\infty \cdot \|\widetilde{\mathbf{z}}_1\|_2 + \sqrt{md} \|\mathbf{F}\|_\infty \cdot \|\widetilde{\mathbf{z}}_2\|_2 \big) + 2\sqrt{md} \|\mathbf{F}\|_\infty \cdot \|\widetilde{\mathbf{z}}\|_2 \\
&\le \sqrt{2} p (n + \sqrt{nm}) d\eta\phi T + 2\sqrt{md}\eta B < q/2,
\end{aligned}$$

where we have $\|\widetilde{\mathbf{z}}\|_2 \le B$ by definition of $\mathcal{F}_B$ (see Section 3.3), $\|\mathbf{F}\|_\infty, \|\mathbf{G}\|_\infty \le \eta$, and the last equation holds from the assumption in the statement. Moreover, we use the fact that for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$, we have $\|\mathbf{a}^\top \mathbf{b}\|_\infty \le \sqrt{n} \|\mathbf{a}\|_\infty \|\mathbf{b}\|_2$. This implies that the equality holds over R, and in particular, $\mathbf{F} \cdot (\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_i + \mathbf{w}) \bmod p = \mathbf{F}\widetilde{\mathbf{z}} \bmod p$. Therefore, if $\|\widetilde{\mathbf{z}}\|_\infty \le B \le (p-1)/2$, we can recover $\widetilde{\mathbf{z}}$ by further inverting it by $\mathbf{F}$. Hence, we are able to extract $\widetilde{\mathbf{z}}$ such that $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ by first computing $\widetilde{\mathbf{v}}$ as in in Line 8 and then computing $\widetilde{\mathbf{z}} = \mathbf{F}^{-1}\widetilde{\mathbf{v}} \bmod p$. □

3.6 Downgrading to Simplified/Classical Extractable LinHC for Tighter Proofs

Simplified extractable LinHC. The two constructions of extractable LinHC protocols in Section 3.4 can be trivially downgraded to satisfy the simplified definition of extractable LinHC presented in Section 3.2. Specifically, we remove the oracle H and define algorithm KeyGen to output random matrices $(\mathbf{A}, \mathbf{B})$ (resp. $\mathbf{H}$) in the first (resp. second) construction without querying the random oracle H. One of the benefits is that the proofs of the simplified version of zero-knowledge and $\mathcal{F}$-almost straight-line extractability defined in Section 3.2 now only rely on the standard MLWE, DSMR, and pseudorandomness of PRF against quantum adversaries. Consequently, this allows for a tighter proof, where recall that the quantum accessible version of MLWE and DSMR do not have a tight reduction from standard MLWE and DSMR since we need to rely on either a naive hybrid argument or on Lemma 2.8.

Classical extractable LinHC. The proofs of our two extractable LinHC protocols in Section 3.4 are almost identical in the classical setting. In the classical setting, the reduction loss of the (non-simplified) zero-knowledge and $\mathcal{F}$-almost straight-line extractability will all be only linearly dependent on the number of random oracle query $\mathcal{A}$ makes. For instance the term $\sqrt{C \cdot Q^3 \cdot \frac{\mathsf{err}}{\mu(\phi,\mathsf{err})}}$ in the statement of zero-knowledge (Lemmata 3.12 and 3.16) becomes $Q \cdot \frac{\mathsf{err}}{\mu(\phi,\mathsf{err})}$. In addition, we no longer require a PRF in the proof of $\mathcal{F}$-almost straight-line extractability since we can lazily sample the random oracle.

4 How to Use Extractable LinHC

In this section, we provide a basic example of bootstrapping the classical ROM secure Lyubashevsky's Σ-protocol [Lyu09, Lyu12] to be QROM secure using an extractable LinHC protocol. The aim of this section is to provide a guide on how to prove QROM security using an extractable LinHC protocol. In Section 5, we see how these ideas can be used to prove QROM security of more complex protocols.

As explained in the beginning of Section 3, we can either construct a (1) quantum straight-line extractable Σ-protocol using the simplified extractable LinHC protocol (see Section 3.2) or a (2) quantum secure simulation straight-line extractable NIZK (or a signature scheme) using the standard extractable LinHC protocol. We explain both items. The former is easier to prove and makes it simpler to understand the essence of the extractable LinHC protocol, while the latter provides a stronger and more useful result.

4.1 Lyubashevsky's Σ-Protocol $\Rightarrow$ Quantum Secure Σ-Protocol via Simplified Extractable LinHC

We show how to make the classical lattice-based Σ-protocol of Lyubashevsky into a Σ-protocol that is quantum straight-line proof of knowledge in the CRS model by integrating it with a simplified extractable LinHC in the standard model. Since the CRS is a random bit string, we can get rid of it in the (Q)ROM by allowing the prover to generate the CRS as an output of the (Q)RO. Recall Lyubashevsky's Σ-protocol allows a prover to prove possession of a (module) short integer solution (MSIS) instance. That is, the prover proves possession of a short vector $\mathbf{e} \in R_q^m$ such that $\mathbf{Ae} = \mathbf{u}$ for public $(\mathbf{A}, \mathbf{u})$.[19] Below, we denote Lyubashevsky's Σ-protocol as Σ_{Lyu}-protocol.

Preparation. Let $\mathsf{ChSet} \subset \{0,1\}^\kappa$ be a set such that all $\beta \in \mathsf{ChSet}$ satisfies $\|\beta\|_1 \leq \ell$. Here, ℓ is chosen in such a way to guarantee $\binom{n}{\ell} \geq 2^{256}$. Let ϕ and err be parameters specified by the rejection sampling algorithm Lemma 2.10. Let $B_{\mathbf{e}}$, $B_{\mathbf{r}}$, and $B_{\mathbf{z}}$ be positive reals such that $B_{\mathbf{r}} \geq \sqrt{2md} \cdot \ell \cdot B_{\mathbf{e}}$ and $B_{\mathbf{z}} \geq \sqrt{2nd} \cdot \phi \cdot B_{\mathbf{r}}$. Define the MSIS relation as $\mathcal{R}_{\mathsf{MSIS}} = \{(\mathsf{X} := (\mathbf{A}, \mathbf{u}), \mathsf{W} := \mathbf{e}) \mid \mathbf{Ae} = \mathbf{u} \ \wedge \ \|\mathbf{e}\|_2 \leq B_{\mathbf{e}}\}$, where $\mathbf{A} \in R_q^{n \times m}$, $\mathbf{u} \in R_q^n$, and $\mathbf{e} \in R_q^m$. We also define the "relaxed" relation $\mathcal{R}'_{\mathsf{MSIS}}$ where the only difference between $\mathcal{R}_{\mathsf{MSIS}}$ is that $\mathbf{e}$ now only satisfies $\mathbf{Ae} = (\beta - \widetilde{\beta}) \cdot \mathbf{u}$ for some $\beta, \widetilde{\beta} \in \mathsf{ChSet}$ and $\|\mathbf{e}\|_2 \leq B'_{\mathbf{e}}$ for a slightly larger bound $B'_{\mathbf{e}} > B_{\mathbf{e}}$. We provide the Lyubashevsky's original Σ-protocol for relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ in Appendix A.2 for reference. It is known that the Σ_{Lyu}-protocol is naHVZK and satisfies relaxed 2-special soundness.

Our quantum secure Σ-protocol. The construction is depicted in Figure 1. Algorithm Setup of the Σ-protocol runs KeyGen of the extractable LinHC protocol. Below, we show correctness, naHVZK, and $\mathsf{SL\text{-}PoK}$ of our Σ-protocol in Figure 1. The first two properties follow almost immediately from the underlying Σ_{Lyu}-protocol and the simplified extractable LinHC protocol.

Lemma 4.1 (Correctness). *Let the Σ_{Lyu}-protocol for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ and the simplified extractable LinHC protocol have correctness error (δ_0, δ_1) and (δ'_0, δ'_1), respectively. Then, our Σ-protocol in the CRS model for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ in Figure 8 has correctness error $(1 - (1 - \delta_0) \cdot (1 - \delta'_0), 1 - (1 - \delta_1)(1 - \delta'_1))$.*

Proof. The correctness error follows from noticing the rejection probability of the underlying Σ-protocol is independent of that of the extractable LinHC. □

Lemma 4.2 (NaHVZK). *Let the Σ_{Lyu}-protocol for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ and the simplified extractable LinHC protocol be ϵ_{zk}-naHVZK and ϵ'_{zk}-naHVZK with zero-knowledge simulators ZKSim_Σ and $\mathsf{ZKSim}_{\mathsf{LinHC}}$, respectively. Then our Σ-protocol in the CRS model for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ in Figure 8 is $(\epsilon_{\mathsf{zk}} + \epsilon'_{\mathsf{zk}})$-$\mathsf{naHVZK}$ with zero-knowledge simulator ZKSim described in Figure 9.*

Proof. Note we only require the simplified naHVZK defined in Section 3.2. We first modify the real transcript and simulate $(\mathsf{com}, \mathsf{op})$ by $\mathsf{ZKSim}_{\mathsf{LinHC}}(\mathsf{K}_{\mathsf{com}}, \beta, \mathbf{z})$, where $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$. Due to ϵ'_{zk}-naHVZK of the extractable LinHC protocol, the transcripts are indistinguishable. We then further modify the transcript and simulate

[19] To be precise, this is an *inhomogeneous* MSIS instance.

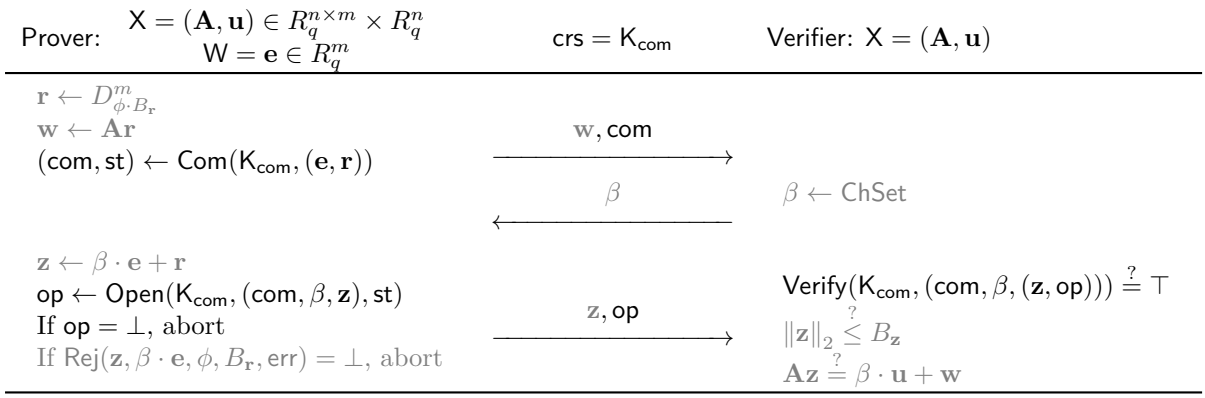

Figure 8: Quantum secure Σ-protocol in the CRS model for the lattice relation $\mathbf{A}\mathbf{e} = \mathbf{u}$, where crs is $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}(1^\kappa)$. The witness $\mathbf{e}$ satisfies $\|\mathbf{e}\|_2 \leq B_{\mathbf{e}}$. The gray indicates the components that are used in the Σ_{Lyu}-protocol. In case abort occurs, we send $\bot$ as the third message.

$(\mathbf{w}, \mathbf{z})$ by $\mathsf{ZKSim}_\Sigma(\mathsf{X}, \beta)$. Due to ϵ_{zk}-naHVZK of the Σ_{Lyu}-protocol, the transcripts are indistinguishable. Since the transcript corresponds to those output by ZKSim, this completes the proof. □

$\underline{\mathsf{ZKSim}(\mathsf{crs} = \mathsf{K}_{\mathsf{com}}, \mathsf{X} = (\mathbf{A}, \mathbf{u}), \beta)}$
1: $(\mathbf{w}, \mathbf{z}) \leftarrow \mathsf{ZKSim}_\Sigma(\mathsf{X}, \beta)$
2: $(\mathsf{com}, \mathsf{op}) \leftarrow \mathsf{ZKSim}_{\mathsf{LinHC}}(\mathsf{K}_{\mathsf{com}}, \beta, \mathbf{z})$
3: **return** $\big((\mathbf{w}, \mathsf{com}), (\mathbf{z}, \mathsf{op})\big)$

$\underline{\mathsf{SimSetup}(1^\kappa)}$
1: $(\widetilde{\mathsf{K}}_{\mathsf{com}}, \tau) \leftarrow \mathsf{SimKeyGen}(1^\kappa)$
2: **return** $(\widetilde{\mathsf{crs}} := \widetilde{\mathsf{K}}_{\mathsf{com}}, \tau)$

$\underline{\mathsf{SL\text{-}Extract}\big(\tau, ((\mathbf{w}, \mathsf{com}), \beta, (\mathbf{z}, \mathsf{op}))\big)}$
1: Run $L \leftarrow \mathsf{LinCExtract}\big(\tau, (\mathsf{com}, \beta, (\mathbf{z}, \mathsf{op}))\big)$ and **return** $\bot$ if does not terminate in time $T^* \cdot \mathsf{poly}(\kappa)$.
2: **if** $L = \bot$ **then return** $\bot$
3: $\{(\beta, \mathbf{z}), (\widetilde{\beta}, \widetilde{\mathbf{z}})\} \leftarrow L$
4: $\mathbf{z}^* \leftarrow \mathsf{Extract}_{\mathsf{ss}}(\mathbf{w}, (\beta, \mathbf{z}), (\widetilde{\beta}, \widetilde{\mathbf{z}}))$
5: **return** $\mathsf{W} := \mathbf{z}^*$

Figure 9: Description of ZKSim, $\mathsf{SimSetup}$, and $\mathsf{SL\text{-}Extract}$ for the Σ-protocol in Figure 8.

Lemma 4.3 ($\mathsf{SL\text{-}PoK}$). *Let the Σ_{Lyu}-protocol for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ be relax 2-special sound with extractor $\mathsf{Extract}_{\mathsf{ss}}$. Let the simplified extractable LinHC protocol be $\epsilon_{\mathsf{IndCom}}$-$\mathcal{F}_{B_{\mathbf{z}}}$-almost straight-line extractable with simulator $\mathsf{SimKeyGen}$ and linear commitment extractor $\mathsf{LinCExtract}$, where $\mathcal{F}_{B_{\mathbf{z}}}$ is the family of functions of the form $f_{\mathbf{A},\mathbf{u},\mathbf{w}}(\beta, \mathbf{z}) = \top$ if and only if $\|\mathbf{z}\|_2 \leq B_{\mathbf{z}}$ and $\mathbf{A}\mathbf{z} = \beta \cdot \mathbf{u} + \mathbf{w}$. Finally, let $T^* = ((\epsilon - \nu_2)/2 - 1/|\mathsf{ChSet}|)^{-1}$ where ϵ is the advantage of the adversary $\mathcal{A}$ and ν_2 is a negligible function as in the statement of Definition 2.5, and $\mathsf{poly}(\kappa)$ is some fixed polynomial independent of $\mathcal{A}$.*

Then our Σ-protocol in the CRS model for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ in Figure 8 is a straight-line PoK with simulator $\mathsf{SimSetup}$ and straight-line extractor $\mathsf{SL\text{-}Extract}$ described in Figure 9.

Proof. Fix any $\mathsf{X} = (\mathbf{A}, \mathbf{u})$. Let $\mathcal{A}$ be a QPT algorithm that outputs a valid transcript with probability ϵ as in the statement of Definition 2.5. Then, by Item 1 of Definition 3.9, we have

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}} = \widetilde{\mathsf{K}}_{\mathsf{com}}, \tau) \leftarrow \mathsf{SimSetup}(1^\kappa) \\ (\alpha, \mathsf{st}) \leftarrow \mathcal{A}(\widetilde{\mathsf{crs}}, \mathsf{X}) \\ \beta \leftarrow \mathsf{ChSet} \\ \gamma \leftarrow \mathcal{A}(\widetilde{\mathsf{crs}}, \mathsf{X}, \alpha, \beta, \mathsf{st})\end{array} : \mathsf{Verify}(\widetilde{\mathsf{crs}}, \mathsf{X}, (\alpha, \beta, \gamma)) = \top\right] \geq \epsilon - \epsilon_{\mathsf{IndCom}}, \tag{1}$$

where $\alpha = (\mathbf{w}, \mathsf{com})$ and $\gamma = (\mathbf{z}, \mathsf{op})$. Let $\Gamma = |\mathsf{ChSet}| \cdot \frac{\epsilon - \epsilon_{\mathsf{IndCom}}}{2}$ which we assume to be a positive integer larger than 2 without loss of generality. Omitting the randomness for simplicity, we can rewrite the left hand

side of Equation (1) as

$$\Pr\left[\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})|\geq\Gamma\right]+\Pr\left[\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})|<\Gamma\right], \tag{2}$$

where, $f\in\mathcal{F}_{B_{\mathbf{z}}}$ is the function that on input $(\beta,\mathbf{z})$, outputs $\top$ if and only if $\|\mathbf{z}\|_2\leq B_{\mathbf{z}}$ and $\mathbf{Az}=\beta\cdot\mathbf{u}+\mathbf{w}$, where $\mathbf{w}$ is the vector included in α output by $\mathcal{A}$. Nota that we use f instead of $f_{\mathbf{A},\mathbf{u},\mathbf{w}}$ as in the statement for better readability. Since β is sampled uniformly random from ChSet and independently of com output by $\mathcal{A}$, and $S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})$ is the set of β's that permit a valid $(\mathbf{z},\mathsf{op})$ we have

$$\Pr\left[\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})|<\Gamma\right]<\frac{\Gamma}{|\mathsf{ChSet}|}=\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}.$$

Combining this with Equations (1) and (2), we have

$$\Pr\left[\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})|\geq\Gamma\right]\geq\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}.$$

Specifically, with probability at least $\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}$, we have $|S_f(\widetilde{\mathsf{K}}_{\mathsf{com}},\mathsf{com})|\geq\Gamma$. Conditioning on such an event, by Item 2 of Definition 3.9, we have that $\mathsf{LinCExtract}(\tau,(\mathsf{com},\beta,(\mathbf{z},\mathsf{op})))$ outputs a tuple $L=\{(\beta,\mathbf{z}),(\widetilde{\beta},\widetilde{\mathbf{z}})\}$ such that $\beta\neq\widetilde{\beta}$ and $f(\widetilde{\beta},\widetilde{\mathbf{z}})=\top$ in time at most $\left(\frac{|\mathsf{ChSet}|}{\Gamma-1}\right)\cdot\mathsf{poly}_{\mathsf{LinHC}}(\kappa)$ with probability at least $1-2^{-\kappa}$, where we set $\delta=\kappa$. By setting $T^*=\frac{|\mathsf{ChSet}|}{\Gamma-1}$ and $\mathsf{poly}(\kappa)=\mathsf{poly}_{\mathsf{LinHC}}(\kappa)$ in Figure 9, with probability at least $\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}\cdot(1-2^{-\kappa})$, $\mathsf{SL\text{-}Extract}$ moves on to Line 3. Be definition of $f\in\mathcal{F}_{B_{\mathbf{z}}}$, $(\mathbf{w},\beta,\mathbf{z})$ and $(\mathbf{w},\widetilde{\beta},\widetilde{\mathbf{z}})$ are two valid transcripts for the underlying classical Σ-protocol. Hence, we obtain $\mathbf{z}^*\leftarrow\mathsf{Extract}_{\mathsf{ss}}(\mathbf{w},(\beta,\mathbf{z}),(\widetilde{\beta},\widetilde{\mathbf{z}}))$ such that $(\mathsf{X},\mathsf{W}=\mathbf{z}^*)\in\mathcal{R}'_{\mathsf{MSIS}}$ as desired.

To summarize, we have

$$\Pr\left[\begin{array}{c}(\widetilde{\mathsf{crs}}=\widetilde{\mathsf{K}}_{\mathsf{com}},\tau)\leftarrow\mathsf{SimSetup}(1^\kappa)\\(\alpha,\mathsf{st})\leftarrow\mathcal{A}(\widetilde{\mathsf{crs}},\mathsf{X})\\\beta\leftarrow\mathsf{ChSet}\\\gamma\leftarrow\mathcal{A}(\widetilde{\mathsf{crs}},\mathsf{X},\alpha,\beta,\mathsf{st})\end{array}:\begin{array}{c}\mathsf{Verify}(\widetilde{\mathsf{crs}},\mathsf{X},(\alpha,\beta,\gamma))=\top\\\mathsf{W}\leftarrow\mathsf{SL\text{-}Extract}(\tau,(\alpha,\beta,\gamma))\\(\mathsf{X},\mathsf{W})\in\mathcal{R}'_{\mathsf{MSIS}}\end{array}\right]\geq\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}(1-2^{-\kappa}),$$

and the runtime of $\mathsf{SL\text{-}Extract}$ is upper bounded by $\left(\frac{\epsilon-\epsilon_{\mathsf{IndCom}}}{2}-\frac{1}{|\mathsf{ChSet}|}\right)^{-1}\cdot\mathsf{poly}_{\mathsf{LinHC}}(\kappa)$. This completes the proof. □

Remark 4.4 (Modulus used by the extractable LinHC). We note that the modulus q used by the extractable LinHC protocol and the underlying Σ_{Lyu}-protocol do not need to be the same.

Remark 4.5 (Transforming quantum secure Σ protocols in the CRS model into NIZKs/signatures). A Σ-protocol in the CRS model with a quantum straight-line proof of knowledge is also a standard quantum proof of knowledge. Therefore, using resent works [LZ19, DFMS19], we can transform our quantum secure Σ-protocol into NIZKs or signatures. However, in the next section, we provide a conceptually simpler proof that converts a non-quantum secure Σ-protocol directly into NIZKs or signatures. Our proof works identically for multi-round public-coin interactive proofs and it provides tighter reduction compared to [LZ19, DFMS19, DFM20] since it does not have to rewind the adversary.

4.2 Lyubashevsky's Σ-Protocol ⇒ QROM Secure Signature via Extractable LinHC and Fiat-Shamir

We show how to directly compile the Σ_{Lyu}-protocol into an $\mathsf{eu\text{-}cma}$ secure signature scheme using the Fiat-Shamir transform (see Appendix A.3 for a definition of signature schemes.) At a high-level the Fiat-Shamir transform compiles a public-coin HVZK interactive protocol into a simulation extractable (tagged) NIZK.

Roughly, this can be thought of as a particular type of signature scheme by viewing the message as the tag. Therefore the main technicality of this section is to show that even if an adversary can observe polynomially many simulated proofs, we are still able to extract a witness from a valid proof output by the adversary *without* rewinding. In the context of signature schemes the (simulated) proofs correspond to (simulated) signatures the adversary observes, and the witness extraction from a forged proof corresponds to extracting the secret key of the signature scheme.

Below, we describe a signature scheme based on the Fiat-Shamir transform rather than a simulation straight-line extractable NIZK. This is a common approach taken in prior works such as [LZ19, DFMS19]. There are mainly two reasons for this: defining such NIZK in the (Q)ROM is more complicated compared to defining a standard signature scheme, and a typical proof of a signature scheme based on the Fiat-Shamir transform captures all the essence of such NIZK. Specifically, the proof implicitly constructs a reduction algorithm that simulates a NIZK proof to the adversary and extracts the witness from the forgery output by the adversary. Therefore, in this paper we view (straight-line) extractable NIZKs as a signature scheme, and vice versa, whenever the context is clear.

Finally, as we explained in the introduction, we do not provide any concrete parameters for our resulting signature scheme since we believe it would be less efficient compared to QROM secure Dilithium proposed in [KLS18]. Roughy, this is because when viewed as an NIZK, [KLS18] only achieves soundness, where ours achieve a stronger proof of knowledge (i.e., there is an extractor that extracts the witness from a forged proof). Hence, keep in mind that the main focus of this section is to provide a tutorial on how one would use an extractable LinHC in a security proof.

Our QROM secure signature scheme. The construction of our (deterministic) signature scheme in the QROM is provided in Figure 10.[20] The algorithms are provided oracle access to the random oracle H, and we use appropriate domain separation to simulate two independent random oracles with different domains and ranges: $\mathsf{H}_{\mathtt{LHC}}$ for the extractable LinHC protocol and $\mathsf{H}_{\mathtt{FS}}$ for applying the Fiat-Shamir transform (see Figure 11). The output space of $\mathsf{H}_{\mathtt{FS}}$ is $\mathsf{ChSet} := \{\beta \in \{0,1\}^\kappa \mid \|\beta\|_1 \leq \ell\}$. Let all the parameters be defined identically to those of the Σ-protocol. We assume that each first message ($\mathbf{w} = \mathbf{Ar}$) of the underlying Σ_{Lyu}-protocol has ζ-min-entropy (see Definition 2.6), and further assume with overwhelming probability that there exists at least two short vectors $\mathbf{e}, \mathbf{e}' \in S^m_{B_{\mathbf{e}}}$ such that $\mathbf{Ae} = \mathbf{Ae}' = \mathbf{u}$. Both of these assumptions are standard in prior works.

Properties. Correctness can be checked similarly to Lemma 4.1. We provide the proof of eu-cma security.

Lemma 4.6 (Eu-cma security). *Let the extractable* LinHC *protocol satisfy* ϵ_{zk}*-*QAnaHVZK *and* ϵ_{IndO}*-*$\mathcal{F}_{B_{\mathbf{z}}}$*-almost straight line extractability, where the function family* $\mathcal{F}_{B_{\mathbf{z}}}$ *is defined identically to Lemma 4.3. Then, assuming the hardness of the* $\mathsf{MSIS}_{n,m,B}$ *problem for* $B = 2 \cdot B_{\mathbf{z}} + B_{\mathbf{r}}$ *and the quantum accessible pseudorandomness of the* PRF*, the signature scheme in Figure 10 is* eu-cma *secure.*

Proof. Let $\mathcal{A}$ be a quantum adversary against the eu-cma security of the signature scheme with advantage ϵ. Assume $\mathcal{A}$ issues at most $Q_{\mathtt{LHC}}$, $Q_{\mathtt{FS}}$, and Q_{S} queries to $\mathsf{H}_{\mathsf{LinHC}}$, $\mathsf{H}_{\mathtt{FS}}$, and the signing oracle, respectively. Below, we consider a sequence of games and denote E_i the event that $\mathcal{A}$ breaks the eu-cma security in Game_i. We denote by S^{msg} the set of messages queried to the signing oracle.

Game_0 : This is the real eu-cma security game. By assumption, we have $\Pr[\mathsf{E}_0] = \epsilon$.

Game_1 : In this game, the challenger modifies the description of $\mathsf{H}_{\mathtt{FS}}$ as in Figure 11. Here, RF used within GetTrans is a uniformly random function with an appropriate domain and range.[21] Since β is uniform random regardless of the input to $\mathsf{H}_{\mathtt{FS}}$, the distribution of these oracles are identical. Therefore, we have

$$\Pr[\mathsf{E}_0] = \Pr[\mathsf{E}_1].$$

[20] Strictly speaking, we require an upper bound on the number of loops we perform in the **while** clause to make the signature algorithm terminate in strict polynomial time. However, since our main focus is to showcase how to use the extractable LinHC protocol and this issue can be handled in a straightforward manner (see [KLS18] for example), we ignore this unrelated subtlety for better readability.

[21] As in Footnote 20, we intentionally ignore the fact that GetTrans only runs in expected polynomial time without loss of generality.

$\underline{\mathsf{S.KeyGen}^{\mathsf{H}}(1^\kappa)}$

1: $(\mathbf{A}, \mathbf{e}) \leftarrow R_q^{n\times m} \times S_{B_\mathbf{e}}^m$
2: $\mathbf{u} = \mathbf{Ae}$
3: $\mathsf{K} \leftarrow \mathcal{K}$
4: $\mathsf{vk} := (\mathbf{A}, \mathbf{u})$
5: $\mathsf{sk} := (\mathbf{e}, \mathsf{K})$
6: **return** $(\mathsf{vk}, \mathsf{sk})$

$\underline{\mathsf{S.Verify}^{\mathsf{H}}(\mathsf{vk}, \sigma, \mathsf{M})}$

1: $(\beta, \mathbf{z}, \mathsf{com}, \mathsf{op}) \leftarrow \sigma$
2: $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}]$
3: $b \leftarrow \mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \beta, (\mathbf{z}, \mathsf{op})))$
4: **if** $b = \bot$ **then return** $\bot$
5: $\mathbf{w} \leftarrow \mathbf{Az} - \beta \cdot \mathbf{u}$
6: **if** $\|\mathbf{z}\|_2 > B_\mathbf{z}$ or $\beta \neq \mathsf{H}_{\mathtt{FS}}(\mathbf{w}\|\mathsf{com}\|\mathsf{M})$ **then return** $\bot$
7: **else return** $\top$

$\underline{\mathsf{S.Sign}^{\mathsf{H}}(\mathsf{vk}, \mathsf{sk}, \mathsf{M})}$

1: $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}]$
2: $(b, \mathsf{op}, c) \leftarrow (\bot, \bot, 0)$
3: **while** $b = \bot \vee \mathsf{op} = \bot$ **do**
4: $\quad \rho_\mathbf{r} \| \rho_{\mathsf{Rej}} \| \rho_{\mathsf{Com}} \| \rho_{\mathsf{Open}} \leftarrow \mathsf{PRF}(\mathsf{K}, \mathsf{M}\|c)$
5: $\quad \mathbf{r} \leftarrow D_{\phi\cdot B_\mathbf{r}}^m[\rho_\mathbf{r}]$
6: $\quad \mathbf{w} \leftarrow \mathbf{Ar}$
7: $\quad (\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}, \mathbf{r}))[\rho_{\mathsf{Com}}]$
8: $\quad \beta \leftarrow \mathsf{H}_{\mathtt{FS}}(\mathbf{w}\|\mathsf{com}\|\mathsf{M})$
9: $\quad \mathbf{z} \leftarrow \beta \cdot \mathbf{e} + \mathbf{r}$
10: $\quad b \leftarrow \mathsf{Rej}(\mathbf{z}, \beta \cdot \mathbf{e}, \phi, B_\mathbf{r}, \mathsf{err})[\rho_{\mathsf{Rej}}]$
11: $\quad \mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \beta, \mathbf{z}), \mathsf{st})[\rho_{\mathsf{Open}}]$
12: $\quad c \leftarrow c + 1$
13: **return** $\sigma := (\beta, \mathbf{z}, \mathsf{com}, \mathsf{op})$

Figure 10: QROM secure signature scheme by applying the Fiat-Shamir transform to our Σ-protocol in Figure 8. Oracles $\mathsf{H}_{\mathtt{LHC}}$ and $\mathsf{H}_{\mathtt{FS}}$ are defined in Figure 11.

Game_2 : In this game, the challenger uses a modified signing algorithm $\mathsf{S.SimSign}$ depicted in Figure 11 to answer the signing query. Since the hash function $\mathsf{H}_{\mathtt{FS}}$ is patched in Game_1, the only difference between the previous game comes from the difference in using a PRF. Due to the quantum accessible pseudorandomness of the PRF, for a QPT algorithm $\mathcal{B}_{1,\mathsf{PRF}}$, we have

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \mathsf{Adv}^{\mathsf{PRF}}(\mathcal{B}_{1,\mathsf{PRF}}).^{22}$$

Game_3 : In this game, the challenger adds a winning condition to the forgery output by the adversary. This is depicted in the $\mathsf{CheckWinCond}$ algorithm in Figure 11. Concretely, the challenger additionally checks whether $\mathbf{w}^* = \mathbf{Az}^* - \beta^* \cdot \mathbf{u}$ is equal to that output by $\mathsf{GetTrans}(\mathsf{M}^*)$. If so it aborts the game, and otherwise, it is defined exactly as in the previous game. By our assumption, each first message $\mathbf{w}$ of the underlying Σ-protocol has ζ-min-entropy if M^* was never queried to the signing oracle, i.e., $\mathsf{M}^* \notin S^{\mathsf{msg}}$. Therefore, we have

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq 2^{-\zeta}.$$

Game_4 : In this game, the challenger modifies the description of $\mathsf{GetTrans}$ used within $\mathsf{H}_{\mathtt{FS}}$ and $\mathsf{S.SimSign}$ as defined in Figure 11. We show that the two games are indistinguishable assuming that the extractable LinHC protocol is $\mathsf{QAnaHVZK}$. Concretely, fix $\mathbf{e}$ and let $D_{\beta,\mathbf{r}}$ be a distribution that samples uniformly random $\beta \leftarrow \mathsf{ChSet}$ and $\mathbf{r} \leftarrow D_{\phi\cdot B_\mathbf{z}}^m$ until $\top \leftarrow \mathsf{Rej}(\beta \cdot \mathbf{e} + \mathbf{r}, \beta \cdot \mathbf{e}, \phi, B_\mathbf{z}, \mathsf{err})$, that finally outputs $(\beta, \mathbf{r})$. Then, by viewing M as the randomness ρ in Definition 3.3, we see that $\mathsf{GetTrans}(\mathsf{M})$ in Game_3 (resp. Game_4) can be simulated by running $D_{\mathsf{trans}}^{\not\perp}(\mathsf{M}, \mathbf{e})$ (resp. $D_{\mathsf{sim}}(\mathsf{M}, \mathbf{e})$). Here, notice that since $\mathbf{r}$ is provided as output of $D_{\mathsf{trans}}^{\not\perp}(\mathsf{M}, \mathbf{e})$ and $D_{\mathsf{sim}}(\mathsf{M}, \mathbf{e})$), we can simulate $\mathbf{w} = \mathbf{Ar}$. Therefore, we can construct an adversary $\mathcal{B}_{\mathsf{QAnaHVZK}}$ that makes at most $Q_{\mathtt{LHC}}$ quantum queries to $\mathsf{H}_{\mathtt{LHC}}$ and $Q_{\mathtt{FS}} + Q_{\mathsf{S}}$

[22] A keen reader may have noticed that the QPT $\mathcal{B}_{1,\mathsf{PRF}}$ must efficiently simulate the random oracle H. Here, we implicitly rely on the fact that this can be simulated unconditionally by a $2Q_{\mathtt{FS}}$-wise independent function $f_{2Q_{\mathtt{FS}}}$ in time roughly to evaluate $f_{2Q_{\mathtt{FS}}}$ $Q_{\mathtt{FS}}$-times [Zha12b]. We implicitly rely on this throughout the proof and omit it for simplicity.

$\underline{\mathsf{H}_{\mathtt{LHC}}(\mathsf{M})}$
1: **return** $\mathsf{H}(\mathtt{LHC}\|\mathsf{M})$ ▷ G_0-G_6
2: **return** $\widetilde{\mathsf{H}}_{\mathsf{LinHC}}(\mathsf{M})$ ▷ G_7

$\underline{\mathsf{H}_{\mathtt{FS}}(\mathbf{w}\|\mathsf{com}\|\mathsf{M})}$
1: $(\beta_{\mathsf{M}}, \mathbf{w}_{\mathsf{M}}, \mathbf{z}_{\mathsf{M}}, \mathsf{com}_{\mathsf{M}}, \mathsf{op}_{\mathsf{M}}) \leftarrow \mathsf{GetTrans}(\mathsf{M})$ ▷ G_1-G_7
2: **if** $(\mathbf{w}, \mathsf{com}) = (\mathbf{w}_{\mathsf{M}}, \mathsf{com}_{\mathsf{M}})$ **then return** β_{M} ▷ G_1-G_7
3: **return** $\mathsf{H}(\mathtt{FS}\|\mathbf{w}\|\mathsf{com}\|\mathsf{M})$

$\underline{\mathsf{S.SimSign}^{\mathsf{H}}(\mathsf{vk}, \mathsf{sk}, \mathsf{M})}$ // G_2-G_7
1: $S^{\mathsf{msg}} \leftarrow S^{\mathsf{msg}} \cup \{\mathsf{M}\}$
2: $(\beta, \mathbf{w}, \mathbf{z}, \mathsf{com}, \mathsf{op}) \leftarrow \mathsf{GetTrans}(\mathsf{M})$
3: **return** $\sigma := (\beta, \mathbf{z}, \mathsf{com}, \mathsf{op})$

$\underline{\mathsf{GetTrans}(\mathsf{M})}$ // G_1-G_3
1: $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}]$
2: $(b, \mathsf{op}, c) \leftarrow (\bot, \bot, 0)$
3: **while** $b = \bot \vee \mathsf{op} = \bot$ **do**
4: $\quad \beta\|\rho_{\mathbf{r}}\|\rho_{\mathsf{Rej}}\|\rho_{\mathsf{Com}}\|\rho_{\mathsf{Open}} \leftarrow \mathsf{RF}(\mathsf{M}\|c)$
5: $\quad \mathbf{r} \leftarrow D^m_{\phi \cdot B_{\mathbf{z}}}[\rho_{\mathbf{r}}]$
6: $\quad \mathbf{w} \leftarrow \mathbf{Ar}$
7: $\quad (\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}, \mathbf{r}))[\rho_{\mathsf{Com}}]$
8: $\quad \mathbf{z} \leftarrow \beta \cdot \mathbf{e} + \mathbf{r}$
9: $\quad b \leftarrow \mathsf{Rej}(\mathbf{z}, \beta \cdot \mathbf{e}, \phi, B_{\mathbf{z}}, \mathsf{err})[\rho_{\mathsf{Rej}}]$
10: $\quad \mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \beta, \mathbf{z}), \mathsf{st})[\rho_{\mathsf{Open}}]$
11: $\quad c \leftarrow c + 1$
12: **return** $(\beta, \mathbf{w}, \mathbf{z}, \mathsf{com}, \mathsf{op})$

$\underline{\mathsf{GetTrans}(\mathsf{M})}$ // G_4
1: $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}]$
2: $(b, c) \leftarrow (\bot, 0)$
3: **while** $b = \bot$ **do**
4: $\quad \beta\|\rho_{\mathbf{r}}\|\rho_{\mathsf{Rej}}\|\rho_{\mathtt{LHC}} \leftarrow \mathsf{RF}(\mathsf{M}\|c)$
5: $\quad \mathbf{r} \leftarrow D^m_{\phi \cdot B_{\mathbf{z}}}[\rho_{\mathbf{r}}]$
6: $\quad \mathbf{w} \leftarrow \mathbf{Ar}$
7: $\quad \mathbf{z} \leftarrow \beta \cdot \mathbf{e} + \mathbf{r}$
8: $\quad b \leftarrow \mathsf{Rej}(\mathbf{z}, \beta \cdot \mathbf{e}, \phi, B_{\mathbf{z}}, \mathsf{err})[\rho_{\mathsf{Rej}}]$
9: $\quad c \leftarrow c + 1$
10: $(\mathsf{com}, \mathsf{op}) \leftarrow \mathsf{ZKSim}_{\mathsf{LinHC}}(\mathsf{K}_{\mathsf{com}}, \beta, \mathbf{z})[\rho_{\mathtt{LHC}}]$
11: **return** $(\beta, \mathbf{w}, \mathbf{z}, \mathsf{com}, \mathsf{op})$

$\underline{\mathsf{GetTrans}(\mathsf{M})}$ // $\mathsf{G}_5, \mathsf{G}_6$
1: $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}^{\mathsf{H}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}]$
2: $\beta\|\rho_\Sigma\|\rho_{\mathtt{LHC}} \leftarrow \mathsf{RF}(\mathsf{M})$ ▷ G_5
3: $\beta\|\rho_\Sigma\|\rho_{\mathtt{LHC}} \leftarrow \mathsf{PRF}(\mathsf{K}, \mathsf{M})$ ▷ G_6
4: $\mathbf{z} \leftarrow D^m_{\phi \cdot B_{\mathbf{z}}}[\rho_\Sigma]$
5: $\mathbf{w} \leftarrow \mathbf{Az} - \beta \cdot \mathbf{u}$
6: $(\mathsf{com}, \mathsf{op}) \leftarrow \mathsf{ZKSim}_{\mathsf{LinHC}}(\mathsf{K}_{\mathsf{com}}, \beta, \mathbf{z})[\rho_{\mathtt{LHC}}]$
7: **return** $(\beta, \mathbf{w}, \mathbf{z}, \mathsf{com}, \mathsf{op})$

$\underline{\mathsf{CheckWinCond}(\sigma^*, \mathsf{M}^*)}$ // G_3-G_7
1: $w \leftarrow \mathsf{S.Verify}^{\mathsf{H}}(\mathsf{vk}, \sigma^*, \mathsf{M}^*)$
2: **if** $\mathsf{M}^* \in S^{\mathsf{msg}} \vee w = \bot$ **then return** $\bot$
3: $(\beta^*, \mathbf{z}^*, \mathsf{com}^*, \mathsf{op}^*) \leftarrow \sigma^*$
4: $\mathbf{w}^* \leftarrow \mathbf{Az}^* - \beta^* \cdot \mathbf{u}$
5: $(\beta_{\mathsf{M}^*}, \mathbf{w}_{\mathsf{M}^*}, \mathbf{z}_{\mathsf{M}^*}, \mathsf{com}_{\mathsf{M}^*}, \mathsf{op}_{\mathsf{M}^*}) \leftarrow \mathsf{GetTrans}(\mathsf{M}^*)$
6: **if** $\mathbf{w}^* = \mathbf{w}_{\mathsf{M}^*}$ **then return** $\bot$
7: **return** $\top$

Figure 11: Description of oracles, where LHC and FS are special bit strings used nowhere else in the scheme. CheckWinCond checks whether the forgery output by $\mathcal{A}$ satisfies the winning condition in each Game_i. G_i is an abbreviation for Game_i and RF denotes a random function defined over an appropriate domain and range.

queries (out of which Q_{S} is classical) to oracles $D^{\mathcal{V}}_{\mathsf{trans}}$ or D_{sim} such that

$$|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \leq \mathsf{Adv}^{\mathsf{QAnaHVZK}}(\mathcal{B}_{\mathsf{QAnaHVZK}}).$$

Game_5 : In this game, the challenger further modifies the description of GetTrans used within $\mathsf{H}_{\mathtt{FS}}$ and S.SimSign as defined in Figure 11. Due to rejection sampling (Lemma 2.10), each signature returned by the signing oracle is distributed $\frac{\mathsf{err}}{\mu(\phi, \mathsf{err})}$-close to the previous game. Moreover, the output distribution of oracle $\mathsf{H}_{\mathtt{FS}}$ remains the same since in both games it outputs a random challenge regardless of $\mathbf{w} = \mathbf{w}_{\mathsf{M}}$. Therefore, we have

$$|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \leq Q_{\mathsf{S}} \cdot \frac{\mathsf{err}}{\mu(\phi, \mathsf{err})}.$$

Game_6 : In this game, the challenger makes a final modification to the description of GetTrans used within $\mathsf{H}_{\mathtt{FS}}$ and S.SimSign as defined in Figure 11. That is, GetTrans derives the randomness from a PRF rather than a truly random function RF. Due to the quantum accessible pseudorandomness of the PRF, for a QPT algorithm $\mathcal{B}_{2,\mathsf{PRF}}$, we have

$$|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \leq \mathsf{Adv}^{\mathsf{PRF}}(\mathcal{B}_{2,\mathsf{PRF}}).$$

Game_7 : In this game, the challenger modifies the description of $\mathsf{H}_{\mathtt{LHC}}$. The challenger first runs the oracle simulator $(\widetilde{\mathsf{H}}_{\mathsf{LinHC}}, \tau) \leftarrow \mathsf{SimOracle}_{\mathsf{LinHC}}(1^\kappa)$ defined by the LinHC protocol. Here, we assume the domain and range of $\widetilde{\mathsf{H}}_{\mathsf{LinHC}}(\cdot)$ is identical to those of $\mathsf{H}(\mathsf{LinHC}\|\cdot)$. Then, the challenger defines $\mathsf{H}_{\mathtt{LHC}}$ as in Figure 11. By the ϵ_{IndO}-$\mathcal{F}_{B_{\mathbf{z}}}$-almost straight line extractability of LinHC, for a QPT algorithm $\mathcal{B}_{\mathsf{IndO}}$ making at most $Q_{\mathtt{LHC}}$ query, we have

$$|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]| \leq \mathsf{Adv}^{\mathsf{IndO}}(\mathcal{B}_{\mathsf{IndO}}).$$

Let $\epsilon^* := \Pr[\mathsf{E}_7]$. Below, in Lemma 4.7 we show we can use $\mathcal{A}$ with a non-negligible advantage ϵ^* to solve the MSIS problem in quantum polynomial time with probability at least $\epsilon^*/4$. Collecting all the bounds, we conclude that $\mathcal{A}$'s advantage ϵ in the original game must be negligible. The following proof of Lemma 4.7 completes the proof of the statement.

Lemma 4.7. *Let $\mathcal{A}$ be a QPT adversary with advantage ϵ^* in Game_7 making at most $Q_{\mathtt{FS}}$ query to oracle $\mathsf{H}_{\mathtt{FS}}$. Then, there exists an adversary $\mathcal{B}_{\mathsf{MSIS}}$ against the $\mathsf{MSIS}_{n,m,B}$ problem for $B = 2 \cdot B_{\mathbf{z}} + B_{\mathbf{r}}$ such that*

$$\mathsf{Adv}^{\mathsf{eu\text{-}cma}}_{\mathsf{Game}_7}(\mathcal{A}) \leq 4 \cdot \mathsf{Adv}^{\mathsf{MSIS}_{n,m,B}}(\mathcal{B}_{\mathsf{MSIS}}) \cdot (1 + 2^{-\kappa+1}),$$

where $\mathsf{Time}(\mathcal{B}_{\mathsf{MSIS}}) = \mathsf{Time}(\mathcal{A}) + \left(\frac{\epsilon^*}{16\cdot(Q_{\mathtt{FS}}+1)^2} - \frac{1}{|\mathsf{ChSet}|}\right)^{-1} \cdot \mathsf{poly}(\kappa)$ for a polynomial $\mathsf{poly}(\kappa)$ independent of $\mathcal{A}$.

Proof. Assume $\mathcal{B}_{\mathsf{MSIS}}$ is given $\mathbf{A} \in R_q^{n\times m}$ as the $\mathsf{MSIS}_{n,m,B}$ problem. To simulate the view of $\mathcal{A}$ in Game_7, $\mathcal{B}_{\mathsf{MSIS}}$ samples $\mathbf{e} \leftarrow S^m_{B_{\mathbf{e}}}$ and provides $\mathcal{A}$ with $\mathsf{vk} = (\mathbf{A}, \mathbf{u} = \mathbf{A}\mathbf{e})$. Due to the modification we made, $\mathcal{B}_{\mathsf{MSIS}}$ is able to simulate the Game_7-challenger without using $\mathbf{e}$. Finally, at some point $\mathcal{A}$ outputs a valid forgery $(\mathsf{M}^*, \sigma^* = (\beta^*, \mathbf{z}^*, \mathsf{com}^*, \mathsf{op}^*))$ with non-negligible probability ϵ^* such that $\mathsf{CheckWinCond}(\sigma^*, \mathsf{M}^*) = \top$. $\mathcal{B}_{\mathsf{MSIS}}$ then sets $\mathsf{trans}^* = (\mathsf{com}^*, \beta^*, (\mathbf{z}^*, \mathsf{op}^*))$ and runs $\mathsf{LinCExtract}(\tau, \mathsf{M}^*, \mathsf{trans}^*)$ of the extractable LinHC protocol. If $\mathsf{LinCExtract}$ does not terminate in time $(\frac{\epsilon^*}{16\cdot(Q_{\mathtt{FS}}+1)^2} - \frac{1}{|\mathsf{ChSet}|})^{-1} \cdot \mathsf{poly}(\kappa)$, it aborts. Here, $\mathsf{poly}(\kappa)$ is the polynomial in Definition 3.4. Moreover, if $\mathsf{LinCExtract}$ terminates but outputs $\bot$, then it also aborts. Otherwise, if $\mathsf{LinCExtract}$ outputs the set $L = \{(\beta^*, \mathbf{z}^*), (\widetilde{\beta}, \widetilde{\mathbf{z}})\}$, then $\mathcal{B}_{\mathsf{MSIS}}$ outputs $\mathbf{v} = (\mathbf{z}^* - \widetilde{\mathbf{z}}) - (\beta^* - \widetilde{\beta}) \cdot \mathbf{e}$ as its solution to the MSIS problem. Let us analyze $\mathcal{B}_{\mathsf{MSIS}}$ below.

Let $\mathbf{w}^* = \mathbf{A}\mathbf{z}^* - \beta^* \cdot \mathbf{u}$ and $\mathsf{K}^*_{\mathsf{com}} = \mathsf{KeyGen}^{\widetilde{\mathsf{H}}_{\mathtt{LHC}}}(1^\kappa)[\mathsf{M}^*]$. Then, we have $\mathsf{Verify}(\mathsf{K}^*_{\mathsf{com}}, (\mathsf{com}^*, \beta^*, (\mathbf{z}^*, \mathsf{op}^*))) = \top$, $\|\mathbf{z}^*\|_2 \leq B_{\mathbf{z}}$, and $\beta^* = \mathsf{H}(\mathtt{FS}\|\mathbf{w}^*\|\mathsf{com}^*\|\mathsf{M}^*)$, where the last equality is guaranteed by $\mathbf{w}^* \neq \mathbf{w}_{\mathsf{M}^*}$. Recall the function $f_{\mathbf{A},\mathbf{u},\mathbf{w}} \in \mathcal{F}_{B_{\mathbf{z}}}$ defined as $f_{\mathbf{A},\mathbf{u},\mathbf{w}}(\beta, \mathbf{z}) = \top$ if and only if $\mathbf{A}\mathbf{z} = \beta \cdot \mathbf{u} + \mathbf{w}$ and $\|\mathbf{z}\|_2 \leq B_{\mathbf{z}}$. Let the set $S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}^*}}(\mathsf{M}^*, \mathsf{com}^*)$ for $(\mathbf{w}^*, \mathsf{com}^*, \mathsf{M}^*)$ output by $\mathcal{A}$ have cardinality larger than $|\mathsf{ChSet}| \cdot \epsilon_{\mathsf{LinHC}}$ for some $\epsilon_{\mathsf{LinHC}} > 0$. Then, by setting $\delta = \kappa$ in Definition 3.4, $\mathsf{LinCExtract}(\tau, \mathsf{M}^*, \mathsf{trans}^*)$ outputs $L = \{(\beta^*, \mathbf{z}^*), (\widetilde{\beta}, \widetilde{\mathbf{z}})\}$ such that $\beta^* \neq \widetilde{\beta}$ and $f_{\mathbf{A},\mathbf{u},\mathbf{w}^*}(\widetilde{\beta}, \widetilde{\mathbf{z}}) = \top$, where $\mathsf{trans}^* = (\mathsf{com}^*, \beta^*, (\mathbf{z}^*, \mathsf{op}^*))$ in time at most $(\epsilon_{\mathsf{LinHC}} - \frac{1}{|\mathsf{ChSet}|})^{-1} \cdot \mathsf{poly}(\kappa)$ with probability at least $1 - 2^{-\kappa}$. We show in Lemma 4.8 below, that $\epsilon_{\mathsf{LinHC}} \geq \frac{\epsilon^*}{16\cdot(Q_{\mathtt{FS}}+1)^2}$ with probability at least $1/2$. This implies that $\mathsf{LinCExtract}$ terminates by outputting the set L with probability at least $\frac{1}{2} \cdot (1 - 2^{-\kappa})$. So as not to interrupt the proof of Lemma 4.7, we postpone its proof to the end. Below, we condition on the event that $L \leftarrow \mathsf{LinCExtract}(\tau, \mathsf{M}^*, \mathsf{trans}^*)$ and $L \neq \bot$. Since $\mathbf{w}^* = \mathbf{A}\mathbf{z}^* - \beta^* \cdot \mathbf{u}$ and $\mathbf{w}^* = \mathbf{A}\widetilde{\mathbf{z}} - \widetilde{\beta} \cdot \mathbf{u}$, we have $\mathbf{A}(\mathbf{z}^* - \widetilde{\mathbf{z}}) = (\beta^* - \widetilde{\beta}) \cdot \mathbf{u}$. Plugging in $\mathbf{A}\mathbf{e} = \mathbf{u}$, we have $\mathbf{A}\mathbf{v} = \mathbf{0}$, where $\mathbf{v} = (\mathbf{z}^* - \widetilde{\mathbf{z}}) - (\beta^* - \widetilde{\beta}) \cdot \mathbf{e}$. Here $\|\mathbf{v}\|_2 \leq 2B_{\mathbf{z}} + \sqrt{md\ell}B_{\mathbf{e}} \leq 2B_{\mathbf{z}} + B_{\mathbf{r}} = B$, where the first inequality follows since we have $\|\beta\|_\infty = 1$ and $\|\beta\|_1 \leq \ell$ for all $\beta \in \mathsf{ChSet}$. Finally, by the assumption that the set $\{\mathbf{e}|\mathbf{A}\mathbf{e} = \mathbf{u}\}$ has cardinality more than 2 for any $\mathbf{u}$ with overwhelming probability over the random choice of $\mathbf{A}$, we conclude that $\mathbf{v} \neq \mathbf{0}$ with at least probability $1/2$ since $\mathbf{e}$ is statistically hidden from $\mathcal{A}$. Hence, $\mathcal{B}_{\mathsf{MSIS}}$ solves the $\mathsf{MSIS}_{n,m,B}$ problem in time $\mathsf{Time}(\mathcal{A}) + (\frac{\epsilon^*}{16\cdot(Q_{\mathtt{FS}}+1)^2} - \frac{1}{|\mathsf{ChSet}|})^{-1} \cdot \mathsf{poly}(\kappa)$ with probability at least $\frac{\epsilon^*}{4} \cdot (1 - 2^{-\kappa})$ as desired. □

Lemma 4.8. *The probability that $\left|S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}^*}}(\mathsf{M}^*, \mathsf{com}^*)\right| < \frac{\epsilon^*}{16\cdot(Q_{\mathtt{FS}}+1)^2} \cdot |\mathsf{ChSet}|$ is at most $\frac{1}{2}$ conditioning on $\mathcal{A}$ winning in Game_7.*

Proof. Fix an arbitrary $\mathsf{vk} = (\mathbf{A}, \mathbf{u})$. Let X be the set of all possible choice of $(\mathbf{w}, \mathsf{com}, \mathsf{M})$ and define $\lambda_x := \frac{|S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}}}(\mathsf{M},\mathsf{com})|}{|\mathsf{ChSet}|}$ for all $x = (\mathbf{w}, \mathsf{com}, \mathsf{M}) \in X$. Moreover, define the set $X_{\mathsf{Good}} \subseteq X$ to be the maximal set such that for any $x \in X_{\mathsf{Good}}$, $\lambda_x \leq \lambda$ where $\lambda := \frac{\epsilon^*}{16\cdot(Q_{\mathsf{FS}}+1)^2}$. Let us construct a quantum algorithm $\mathcal{B}_{\mathsf{GSBP}}$ against the GSBP game that internally runs $\mathcal{A}$. At the outset of the game, $\mathcal{B}_{\mathsf{GSBP}}$ prepares the set of reals $\{\lambda'_x\}_{x\in X}$, where $\lambda'_x = \lambda_x$ if $x \in X_{\mathsf{Good}}$ and $\lambda'_x = 0$ if $x \in X \backslash X_{\mathsf{Good}}$. After $\mathcal{B}_{\mathsf{GSBP}}$ submits the set $\{\lambda'_x\}_{x\in X}$ to the challenger, it is given oracle access to G. It then samples three random functions $\mathsf{RF}_0, \mathsf{RF}_1, \mathsf{RF}_2 \leftarrow \mathsf{Func}(X, \mathsf{ChSet})$ conditioned on $\mathsf{RF}_0(x) \in \mathsf{ChSet}\backslash S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}}}(\mathsf{M}, \mathsf{com})$ and $\mathsf{RF}_1(x) \in S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}}}(\mathsf{M}, \mathsf{com})$ for all $x = (\mathbf{w}, \mathsf{com}, \mathsf{M}) \in X$. Finally, $\mathcal{B}_{\mathsf{GSBP}}$ simulates Game_7 to $\mathcal{A}$ by using its oracle G to simulate $\mathsf{H}(\mathtt{FS}\|\cdot)$. Specifically, to simulate an oracle query to $\mathsf{H}(\mathtt{FS}\|x)$, $\mathcal{B}_{\mathsf{GSBP}}$ first checks $x \in X_{\mathsf{Good}}$ and returns $\mathsf{RF}_2(x)$ if not. Otherwise, it returns $\mathsf{RF}_0(x)$ if $0 \leftarrow \mathsf{G}(x)$ and returns $\mathsf{RF}_1(x)$ if $1 \leftarrow \mathsf{G}(x)$. When $\mathcal{A}$ outputs a signature forgery σ^*, $\mathcal{B}_{\mathsf{GSBP}}$ parses it and outputs $x^* := (\mathbf{w}^*, \mathsf{com}^*, \mathsf{M}^*)$. Let us analyze $\mathcal{B}_{\mathsf{GSBP}}$ below.

First of all, it can be checked that $\mathcal{B}_{\mathsf{GSBP}}$ simulates the view of Game_7 perfectly to $\mathcal{A}$ since the output distribution of oracle $\mathsf{H}(\mathtt{FS}\|\cdot)$ is identical to that of Game_7. For the sake of contradiction, let us assume $\mathcal{A}$ outputs a forgery such that $\left|S_{f_{\mathbf{A},\mathbf{u},\mathbf{w}^*}}(\mathsf{M}^*, \mathsf{com}^*)\right| < \frac{\epsilon^*}{16\cdot(Q_{\mathsf{FS}}+1)^2}\cdot|\mathsf{ChSet}|$ with probability greater than $\frac{\epsilon^*}{2}$. Then, since $x^* \in X_{\mathsf{Good}}$ and $\mathsf{G}(x^*) = 1$ by definition, $\mathcal{B}_{\mathsf{GSBP}}$ winw the GSBP problem with probability greater than $\frac{\epsilon^*}{2}$. However, this is a contradiction to Lemma 2.9 since we must have $\mathsf{Adv}^{\mathsf{GSBP}}(\mathcal{B}_{\mathsf{GSBP}}) \leq 8\cdot\lambda\cdot(Q_{\mathsf{FS}}+1)^2 \leq \frac{\epsilon^*}{2}$. This completes the proof. □

□

Remark 4.9 (Straight-Line Extractable Proofs in the Classical Setting). The proofs provided in the previous sections all translate naturally to the classical setting. In fact, we could go through the same proof while only relying on the simplified extractable LinHC protocol in the classical setting as we can lazily program the random oracle. Moreover, Lemma 4.7 can be proven tighter in the classical setting since a classical adversary $\mathcal{A}$ making Q queries to a random oracle has at most probability $Q/|\mathsf{ChSet}|$ of finding the desired output. Therefore, we would be able to get a tighter reduction compared to the quantum case where the reduction loss would all be linear in Q.

5 Application: Quantum Secure 5-Round Public-Coin Exact Sound Proof and QROM Secure Exact Sound NIZK

In this section, to showcase the generality of the extractable LinHC protocol, we show how to integrate it to the recent 5-round public-coin HVZK interactive *exact sound* proof of Bootle et al [BLS19]. The main motivation for choosing [BLS19] as the case study is because the ideas presented in this section can be directly applied to other recent works [BDL+18, ESLL19, YAZ+19, ALS20]. We can convert the protocol of [BLS19] into either (1) a quantum secure straight-line extractable *interactive* proof using the simplified extractable LinHC protocol (as in Section 4.1) or (2) a quantum secure simulation straight-line extractable NIZK (or a signature scheme) using the extractable LinHC protocol (as in Section 4.2).

5.1 Quantum Secure Exact Sound Interactive Proof via Simplified Extractable LinHC

We first show how to apply the simplified extractable LinHC protocol to Bootle et al's protocol [BLS19] to obtain a 5-round public-coin interactive proof that is quantum secure, straight-line extractable, and exact sound. In brief, Bootle et al. constructs an interactive protocol that allows the prover to prove knowledge of a vector $\mathbf{s} \in \{0, 1, 2\}^d$ satisfying $\mathbf{As} = \mathbf{u}$, where the main difference between Lyubashevsky's protocol is that it it *exact* sound. That is, a knowledge extractor extracts a witness that satisfies the original relation used by the prover (and not a "relaxed" relation). Readers may refer to Appendix B.1 for a minimal introduction on the exact sound interactive protocol of Bootle et al. While zero-knowledge of our protocol is a direct consequence of that of Bootle et al's protocol, soundness needs slightly more work.

Parameters. Following Bootle et al., we chose the dimension d and modulus q so that R_q completely splits into d linear factors modulo q, e.g., d is a power of 2 and $q \equiv 1 \mod 2d$. For a ring element $s \in R_q$, we denote $\hat{\mathbf{s}} \in \mathbb{Z}_q^d$ as the NTT representation of s. Then, for a matrix-vector pair $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{m \times d} \times \mathbb{Z}_q^m$, we consider the relation $cla_{\mathsf{ES}} = \{s \in R_q \mid \mathbf{A}\hat{\mathbf{s}} = \mathbf{u} \ \wedge \ \hat{\mathbf{s}} \in \{0,1,2\}^d\}$. Let C denote the set $\{0, X^i \mid 0 \leq i < 2d\} \subset R_q$, and ϕ and err be parameters specified by the rejection sampling algorithm Lemma 2.10. Let $B_{\mathbf{e}}$, $B_{\mathbf{r}}$, and $B_{\mathbf{z}}$ be positive reals such that $B_{\mathbf{r}} \geq \sqrt{6d} \cdot B_{\mathbf{e}}$ and $B_{\mathbf{z}} \geq \sqrt{12d} \cdot \phi \cdot B_{\mathbf{r}}$, where the size of $B_{\mathbf{e}}$ dictates the hardness of the MLWE assumption.

Our quantum secure exact sound protocol. The protocol is depicted in Figure 12. It can be seen that the way we apply the extractable LinHC protocol is very similar to what was done for Lyubashevsky's protocol (see Figure 8). Here, we could have included another extractable LinHC protocol in the middle to let the prover commit to the witness s and y. Although this would make the proof of straight-line proof of knowledge much easier, we chose not to since this will add a considerable overhead in the concrete proof size since s and y are elements over R_q rather than short elements. Namely, we would require an extractable LinHC protocol that supports a message space R_q and this will be quite costly, where we note that theoretically constructing such protocol is easy using a slight modification of the Regev encryption scheme.

Below, correctness and naHVZK of the Σ-protocol in Figure 12 are straightforward to prove.

Prover: $\mathsf{X} = (\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{m \times d} \times \mathbb{Z}_q^m$, $\mathsf{W} = s \in R_q$	$\mathsf{crs} = (\mathbf{B}, \mathsf{K}_{\mathsf{com}})$	Verifier: $\mathsf{X} = (\mathbf{A}, \mathbf{u})$
$y \leftarrow R_q$ $\mathbf{e} \leftarrow S_{B_{\mathbf{e}}}^6$ $\mathbf{t} \leftarrow \begin{pmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \\ \mathbf{b}_3^\top \\ \mathbf{b}_4^\top \\ \mathbf{b}_5^\top \end{pmatrix} \mathbf{e} + \begin{pmatrix} 0 \\ y \\ s \\ y(2s-3) \\ y^2(s-3) \end{pmatrix} \in R_q^5$	$\xrightarrow{(\mathbf{t}, \mathbf{w}, \mathsf{com})}$	
$\mathbf{w} \leftarrow \mathbf{A}\hat{\mathbf{y}} \in \mathbb{Z}_q^m$ $\mathbf{r} \leftarrow D_{\phi \cdot B_{\mathbf{r}}}^6$ $(\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}, \mathbf{r}))$		
	$\xleftarrow{c}$	$c \leftarrow \mathbb{Z}_q$
$z_0 \leftarrow c \cdot s + y$ $x_0 \leftarrow \mathbf{b}_1^\top \mathbf{r}$ $x_1 \leftarrow (\mathbf{b}_2^\top + c \cdot \mathbf{b}_3^\top)\mathbf{r}$ $x_2 \leftarrow ((z_0 - c)(z_0 - 2c) \cdot \mathbf{b}_3^\top - z_0 \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top)\mathbf{r}$	$\xrightarrow{(z_0, x_0, x_1, x_2)}$	
	$\xleftarrow{\beta}$	$\beta \leftarrow C$
		$\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \beta, (\mathbf{z}, \mathsf{op}))) \stackrel{?}{=} \top$ $\|\mathbf{z}\|_2 \stackrel{?}{\leq} B_{\mathbf{z}}$ $\mathbf{A}\hat{\mathbf{z}}_0 \stackrel{?}{=} c \cdot \mathbf{u} + \mathbf{w}$ $\mathbf{b}_1^\top \mathbf{z} \stackrel{?}{=} \beta \cdot t_1 + x_0$ $(\mathbf{b}_2^\top + c \cdot \mathbf{b}_3^\top)\mathbf{z} + \beta \cdot z_0$ $\stackrel{?}{=} \beta \cdot (c \cdot t_3 + t_2) + x_1$ $((z_0 - c)(z_0 - 2c) \cdot \mathbf{b}_3^\top - z_0 \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top)\mathbf{z}$ $\stackrel{?}{=} \beta \cdot ((z_0 - c)(z_0 - 2c) \cdot t_3 - z_0 \cdot t_4 + t_5) + x_2$
$\mathbf{z} \leftarrow \beta \cdot \mathbf{e} + \mathbf{r}$ $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \beta, \mathbf{z}), \mathsf{st})$ If $\mathsf{op} = \perp$, abort If $\mathsf{Rej}(\mathbf{z}, \beta \cdot \mathbf{e}, \phi, B_{\mathbf{r}}, \mathsf{err}) = \perp$, abort	$\xrightarrow{(\mathbf{z}, \mathsf{op})}$	

Figure 12: Quantum secure exact sound public-coin interactive protocol in the CRS model for the relation $\mathcal{R}_{\mathsf{ES}}$. The witness s satisfies $\mathbf{A}\hat{\mathbf{s}} = \mathbf{u}$ and $\hat{\mathbf{s}} \in \{0,1,2\}^d$. $\mathsf{K}_{\mathsf{com}}$ is the commitment key of the simplified extractable LinHC protocol, $\mathbf{B} \in R_q^{5 \times 6}$ is the public parameter of the (implicit) commitment scheme Π_{Com} (see Appendix A.4), and $\mathbf{b}_i^\top$ denotes its i-th row vector. The gray indicates the components that are used in the protocol of Bootle et al. [BLS19]. In case abort occurs, we send $\perp$ as the fifth message.

Lemma 5.1 (Correctness). *Let Bootle et al's protocol [BLS19] for the relation $\mathcal{R}_{\mathsf{ES}}$ and the simplified extractable* LinHC *protocol have correctness error (δ_0, δ_1) and (δ_0', δ_1'), respectively. Then, our protocol in the CRS model for the relation $\mathcal{R}_{\mathsf{ES}}$ in Figure 12 has correctness error $(1-(1-\delta_0)\cdot(1-\delta_0'), 1-(1-\delta_1)(1-\delta_1'))$.*

Proof. The correctness error follows from noticing the rejection probability of the underlying Bootle et al.'s protocol is independent of that of the extractable LinHC. □

Lemma 5.2 (NaHVZK). *Let Bootle et al's protocol [BLS19] for the relation $\mathcal{R}_{\mathsf{ES}}$ and the simplified extractable* LinHC *protocol be ϵ_{zk}-naHVZK and ϵ'_{zk}-naHVZK with zero-knowledge simulators $\mathsf{ZKSim}_{\mathsf{ES}}$ and $\mathsf{ZKSim}_{\mathsf{LinHC}}$, respectively. Then our protocol in the CRS model for the relation $\mathcal{R}_{\mathsf{ES}}$ in Figure 12 is $(\epsilon_{\mathsf{zk}}+\epsilon'_{\mathsf{zk}})$-naHVZK.*

Proof. The zero-knowledge simulator ZKSim is given random challenges $c \leftarrow \mathbb{Z}_q$ and $\beta \leftarrow C$ as inputs and runs $\big((\mathbf{t},\mathbf{w}),(z_0,x_0,x_1,x_2),\mathbf{z}\big) \leftarrow \mathsf{ZKSim}_{\mathsf{ES}}(\mathsf{crs}_{\mathsf{ES}} = \mathbf{B}, \mathsf{X} = (\mathbf{A},\mathbf{u}),(c,\beta))$ and $(\mathsf{com},\mathsf{op}) \leftarrow \mathsf{ZKSim}_{\mathsf{LinHC}}(\mathsf{K}_{\mathsf{com}},\beta,\mathbf{z})$. It then outputs $\big((\mathbf{t},\mathbf{w},\mathsf{com}),(z_0,x_0,x_1,x_2),(\mathbf{z},\mathsf{op})\big)$ as the transcript. Similarly to the proof of Lemma 4.2, indistinguishability of the transcript output by ZKSim and the real prover follows from a simple hybrid argument where we first invoke $\mathsf{ZKSim}_{\mathsf{LinHC}}$ and then $\mathsf{ZKSim}_{\mathsf{ES}}$. □

The high level idea of the proof for straight-line proof of knowledge is similar to those provided by Bootle et al. [BLS19, Theorem 3.1]. The main difference is how we extract a witness from *partial* valid transcripts. Recall Bootle et al. first rewinds the adversary to obtain six valid transcripts with a specific form and then shows how to extract a witness from such transcripts. In our proof, we are only able to extract a small portion of the six valid transcripts so we need to rely on a different argument compared to Bootle et al.

Lemma 5.3 (SL-PoK). *Let the simplified extractable* LinHC *protocol be $\epsilon_{\mathsf{IndCom}}$-$\mathcal{F}_{B_{\mathbf{z}}}$-almost straight-line extractable with simulator* SimKeyGen *and linear commitment extractor* LinCExtract*, where $\mathcal{F}_{B_{\mathbf{z}}}$ is the singleton set $\{f\}$ for a f such that $f(\beta,\mathbf{z}) = \top$ if and only if $\|\mathbf{z}\|_2 \le B_{\mathbf{z}}$. Moreover, let it satisfy the optional requirements in Definition 3.9, Items 3 and 4.*

Then, there exists a PPT simulator SimSetup *and a straight-line extractor* SL-Extract *with the following property: Let $\mathcal{A}$ be an adversary that outputs a valid transcript with probability $\epsilon > 3/q + 2/d$.[23] Then, on input a valid transcript output by $\mathcal{A}$ executed on a simulated* crs *output by* SimSetup*,* SL-Extract *outputs either a witness $s \in R_q$ in the relation $\mathcal{R}_{\mathsf{ES}}$ or a $\mathsf{MSIS}_{n,6n,8B_{\mathbf{z}}}$ solution for $\mathbf{b}_1^\top$ with probability $\epsilon/3 - \nu$ for a negligible function ν. Moreover, the runtime of* SL-Extract *is independent of the runtime of $\mathcal{A}$ and depends only polynomially on d and $\log q$.*

Proof. Assume $\mathcal{A}$ successfully fools the honest verifier with advantage $\epsilon > 3/q + 2/d$ and the resulting transcript is $\mathsf{trans}^* = \big((\mathbf{t},\mathbf{w},\mathsf{com}), c^{(1)}, (z_0^{(1)}, x_0^{(1)}, x_1^{(1)}, x_2^{(1)}), \beta^{(1,1)}, (\mathbf{z}^{(1,1)}, \mathsf{op}^{(1,1)})\big)$.

We first establish that if $\mathcal{A}$ has advantage greater than $3/q+2/d$, then with probability at least $1/3$, the following property: there exists at least three distinct first challenges $c^{(1)}, c^{(2)}, c^{(3)} \in \mathbb{Z}_q$ and two distinct second challenges $\beta^{(k,1)}, \beta^{(k,2)} \in C$ for each $k \in [3]$ such that there exists some third message $(z_0^{(k)}, x_0^{(k)}, x_1^{(k)}, x_2^{(k)})$ and fifth message $(\mathbf{z}^{(k,j)}, \mathsf{op}^{(k,j)})$ where $\mathsf{trans}^{(k,j)} = \big((\mathbf{t},\mathbf{w},\mathsf{com}), c^{(k)}, (z_0^{(k)}, x_0^{(k)}, x_1^{(k)}, x_2^{(k)}), \beta^{(k,j)}, (\mathbf{z}^{(k,j)}, \mathsf{op}^{(k,j)})\big)$ is a valid transcript for all $(k,j) \in [3]\times[2]$. Let $\mathsf{Verify}_{\mathsf{ES}}$ be the verification algorithm for the interactive protocol and denote $S(\alpha) \subseteq \mathbb{Z}_q \times C$ the set of challenges for which there exists a valid response, where $\alpha = (\mathbf{t},\mathbf{z},\mathsf{com})$ is the first message sent by the adversary. Specifically, $S(\alpha) := \{(c,\beta) | \exists(\gamma_1,\gamma_2)$ s.t. $\mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\alpha,c,\gamma_1,\beta,\gamma_2) = \top\}$. Further define BAD the event that there does not exist distinct $c^{(1)}, c^{(2)}, c^{(3)} \in \mathbb{Z}_q$ such that for some $k \in [3]$, there does not exist distinct $\beta^{(k,1)}, \beta^{(k,2)} \in C$ for which $(c^{(k)}, \beta^{(k,1)}), (c^{(k)}, \beta^{(k,2)}) \in S(\alpha)$. By the definition of event BAD, we must have $|S(\alpha)| \le 2d\cdot 2 + (q-2)$ when BAD occurs. Then, since the first message α is chosen by $\mathcal{A}$ before seeing $(c,\beta) \in \mathbb{Z}_q \times C$, we have $\Pr[\mathsf{BAD}] < 2/q + 1/2d$. Therefore, conditioning on $\mathcal{A}$ succeeding, the probability of $\neg$BAD occurring is lower bounded by

$$\Pr[\neg\mathsf{BAD} | \mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\mathsf{trans}^*) = \top]$$

[23] Bootle et al. [BLS19, Theorem 3.1] only requires $\epsilon > 2/q + 1/d$. Although our proof works in this regime as well, this slight modification makes our proof easier to state and will have minimal impact on the concrete efficiency of the scheme.

$$=\frac{1}{\Pr[\mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\mathsf{trans}^*)=\top]}\cdot\Big(\Pr[\mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\mathsf{trans}^*)=\top]-\Pr[\neg\mathsf{BAD}|\mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\mathsf{trans}^*)=\top]\Big)$$
$$>1-\frac{1}{\Pr[\mathsf{Verify}_{\mathsf{ES}}(\mathsf{crs},\mathsf{X},\mathsf{trans}^*)=\top]}\cdot\Big(\frac{2}{q}+\frac{1}{2d}\Big)>1-\frac{1}{\big(\frac{3}{q}+\frac{2}{d}\big)}\cdot\Big(\frac{2}{q}+\frac{1}{2d}\Big)>\frac{1}{3}.$$

This establishes the bound as desired.

Next, conditioning on $\neg\mathsf{BAD}$ not occurring, we show how $\mathsf{SL\text{-}Extract}$ obtains a list that contains all $((\beta^{(k,j)},\mathbf{z}^{(k,j)}))_{(k,j)\in[3]\times[2]}$ using the straight-line extractability of the simplified extractable LinHC protocol. Let us define $\mathsf{SimSetup}$ to run $(\widetilde{\mathsf{K}}_{\mathsf{com}},\tau)\leftarrow\mathsf{SimKeyGen}(1^\kappa)$ and output $\mathsf{crs}=(\mathbf{B},\widetilde{\mathsf{K}}_{\mathsf{com}})$. Due to the simplified $\epsilon_{\mathsf{IndCom}}$-$\mathcal{F}_{B_{\mathbf{z}}}$-almost straight-line extractability, $\mathcal{A}$ still has advantage $\epsilon/3-\epsilon_{\mathsf{IndCom}}$ in outputting a valid transcript trans^* with the above property run on this modified crs. Next, by Definition 3.9, Item 3, $\mathsf{SL\text{-}Extract}$ can use the extractor of the simplified extractable LinHC protocol $\mathsf{LinCExtract}(\tau,\mathsf{trans}^*)$ to obtain a set $L=((\beta_j,\mathbf{z}_j))_{j\in[d]}$ in time polynomial in $|C|=d$[24], where we are guaranteed to extract all $\beta\in C$ that has a corresponding $(\mathbf{z}',\mathsf{op}')$ such that $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}},(\mathsf{com},\beta,(\mathbf{z}',\mathsf{op}')))=\top$ and $\|\mathbf{z}'\|_2\le B_{\mathbf{z}}$. That is, all the extracted β satisfies $\beta\in S_f(\mathsf{K}_{\mathsf{com}},\mathsf{com})$. Moreover, due to Definition 3.9, Item 4, once com is fixed, there exists at most one $\mathbf{z}'$ satisfying $\mathsf{Verify}(\mathsf{K}_{\mathsf{com}},(\mathsf{com},\beta,(\mathbf{z}',\mathsf{op}')))=\top$ for each $\beta\in C$ and any op' regardless of the choice of the second and third messages (i.e., $c\in\mathbb{Z}_q$ and (z,x_0,x_1,x_2)). Therefore, the extracted $\mathbf{z}$ must be the unique $\mathbf{z}'$. Combining the argument so far, we have established $((\beta^{(k,j)},\mathbf{z}^{(k,j)}))_{(k,j)\in[3]\times[2]}\subseteq L$. Here, note $\beta^{(k,j)}$ and $\beta^{(k',j')}$ may be the same when $k\neq k'$. In the following, we show how $\mathsf{SL\text{-}Extract}$ determines which two tuples $(\beta,\mathbf{z})$ and $(\beta',\mathbf{z}')\in L$ correspond to the tuples $(\beta^{(k,1)},\mathbf{z}^{(k,1)})$ and $(\beta^{(k,2)},\mathbf{z}^{(k,2)})$.

Assume we knew which elements in the set L corresponded to $(\beta^{(k,1)},\mathbf{z}^{(k,1)})$ and $(\beta^{(k,2)},\mathbf{z}^{(k,2)})$ for each $k\in[3]$. Then, since $(\mathsf{trans}^{(k,j)})_{(k,j)\in[3]\times[2]}$ are valid transcripts, we have $\mathbf{b}_1^\top\mathbf{z}^{(k)}=\beta^{(k,j)}\cdot t_1+x_0^{(k)}$ for an unknown $x_0^{(k)}$. By subtracting $j=1,2$ for each $k\in[3]$, we can remove $x_0^{(k)}$ to obtain $\mathbf{b}_1^\top\mathbf{z}^{(k)}-\beta^{(k,1)}\cdot t_1=\mathbf{b}_1^\top\mathbf{z}^{(k)}-\beta^{(k,2)}\cdot t_1$. Notice that we can check this equality with only knowledge of $\mathbf{B}$ in the crs and $\mathbf{t}$ in the first message, which is shared among all the transcripts. With this observation in mind, $\mathsf{SL\text{-}Extract}$ performs the following:

1. Prepare an empty list S and counter $t=1$.
2. For each pair $(\beta,\mathbf{z}),(\beta',\mathbf{z}')\in L$, check if $\mathbf{b}_1^\top\mathbf{z}-\beta\cdot t_1=\mathbf{b}_1^\top\mathbf{z}'-\beta'\cdot t_1$. If not move on to the next pair. Otherwise, add $(t,(\beta.\mathbf{z}),(\beta',\mathbf{z}'))$ to the list S, update $t=t+1$, and move on to the next pair.

For each $(t,(\beta,\mathbf{z}),(\beta',\mathbf{z}'))\in S$, denote $\overline{\beta}_t=\beta-\beta'$ and $\overline{\mathbf{z}}_t=\mathbf{z}-\mathbf{z}'$. Then, we have $\mathbf{b}_1^\top\overline{\mathbf{z}}_t=\overline{\beta}_t\cdot t_1$, which is an approximate solution to the first equation of the commitment $\mathbf{t}$ (see Appendix A.4). Therefore, we can compute openings $\mathsf{M}_{t,2}$, $\mathsf{M}_{t,3}$ and $\mathsf{M}_{t,4}$ and $\mathsf{M}_{t,5}$ of $\mathbf{t}$ by setting $\mathsf{M}_{t,\ell}=t_\ell-\overline{\beta}_t^{-1}\cdot(\mathbf{b}_\ell^\top\overline{\mathbf{z}}_t)\in R_q$ for each $\ell\in\{2,3,4,5\}$. Here, note that these openings are valid relaxed openings for the commitment scheme in Appendix A.4 with $\|\overline{\mathbf{z}}_t\|_2\le 2B_{\mathbf{z}}$. Hence, unless $\mathcal{A}$ breaks the binding property of the commitment (or equivalently the $\mathsf{MSIS}_{n,6n,8B_{\mathbf{z}}}$ problem due to Lemma A.4), we are guaranteed that $\mathsf{M}_{t,2}$, $\mathsf{M}_{t,3}$, $\mathsf{M}_{t,4}$, and $\mathsf{M}_{t,5}$ are the same value for all $t\in|S|$. Conditioning on $\mathcal{A}$ not breaking the $\mathsf{MSIS}_{n,6n,8B_{\mathbf{z}}}$ problem, $\mathsf{SL\text{-}Extract}$ outputs $s^*:=\mathsf{M}_{1,3}=\cdots=\mathsf{M}_{|S|,3}$ as the witness. Here, observe that the runtime of $\mathsf{SL\text{-}Extract}$ is only polynomially related to $|C|=d$: it takes time $d\cdot\mathsf{poly}(\kappa)$ to prepare the list L and takes time at most $d^2\cdot\mathsf{poly}(\kappa)$ to prepare the list S. Therefore, it remains to show that $s^*\in R_q$ output by $\mathsf{SL\text{-}Extract}$ indeed satisfies $\mathbf{A}\hat{\mathbf{s}}^*=\mathbf{u}$ and $\hat{\mathbf{s}}^*\in\{0,1,2\}$, where $\hat{\mathbf{s}}^*\in\mathbb{Z}_q^d$ is the NTT representation of s^*. In the following, since all the messages are the same unless $\mathcal{A}$ breaks the $\mathsf{MSIS}_{n,6n,8B_{\mathbf{z}}}$ problem, we drop the subscript t from the messages M and further denote $y^*=\mathsf{M}_2$.

Although we do not know $(c^{(k)},(z_0^{(k)},x_0^{(k)},x_1^{(k)},x_2^{(k)}))_{k\in[3]}$, we have a list L that is guaranteed to contain $(\beta^{(k,j)},\mathbf{z}^{(k,j)})_{(k,j)\in[3]\times[2]}$ included in $(\mathsf{trans}^{(k,j)})_{(k,j)\in[3]\times[2]}$. For each $(k,j)\in[3]\times[2]$ consider the following verification equation

$$(\mathbf{b}_2^\top+c^{(k)}\cdot\mathbf{b}_3^\top)\mathbf{z}^{(k,j)}+\beta^{(k,j)}\cdot z_0^{(k)}=\beta^{(k,j)}\cdot(c^{(k)}\cdot t_3+t_2)+x_1^{(k)},$$

[24]Since d is the dimension of the lattice, we can assume that it is polynomial in the security parameter κ.

where recall that $z_0^{(k)}$ and $x_1^{(k)}$ are unknown but guaranteed to exist. Subtracting the equations for the same k and $j = 1, 2$, we obtain

$$(\mathbf{b}_2^\top + c^{(k)} \cdot \mathbf{b}_3^\top)\overline{\mathbf{z}}^{(k)} + \overline{\beta}^{(k)} \cdot z_0^{(k)} = \overline{\beta}^{(k)} \cdot (c^{(k)} \cdot t_3 + t_2),$$

where $\overline{\beta}^{(k)} = \beta^{(k,1)} - \beta^{(k,2)}$ and $\overline{\mathbf{z}}^{(k)} = \mathbf{z}^{(k,1)} - \mathbf{z}^{(k,2)}$. Further substituting the commitment openings for t_2 and t_3 to the above equation, we obtain

$$(\mathbf{b}_2^\top + c^{(k)} \cdot \mathbf{b}_3^\top)\overline{\mathbf{z}}^{(k)} + \overline{\beta}^{(k)} \cdot z_0^{(k)} = \overline{\beta}^{(k)} \cdot \Big(c^{(k)} \cdot \big((\overline{\beta}^{(k)})^{-1} \cdot (\mathbf{b}_3^\top \overline{\mathbf{z}}^{(k)}) + s^*\big) + \big((\overline{\beta}^{(k)})^{-1} \cdot (\mathbf{b}_2^\top \overline{\mathbf{z}}^{(k)}) + y^*\big)\Big).$$

Routine calculation shows $z_0^{(k)} = y^* + c^{(k)} \cdot s^*$. By performing the same argument on the final verification equation and substituting the commitment openings for t_4 and t_5, we obtain

$$\Big((y^*)^2 s^* - y^* \mathsf{M}_4 + \mathsf{M}_5\Big) + \Big((y^*(2s^* - 3) - \mathsf{M}_4)s^*\Big) \cdot c^{(k)} + \Big(s^*(s^* - 1)(s^* - 2)\Big) \cdot (c^{(k)})^2 = 0.$$

Since this equation holds for all $k \in [3]$ and $c^{(1)} \neq c^{(2)} \neq c^{(3)} \in \mathbb{Z}_q$, we must have $s^*(s^* - 1)(s^* - 2) = 0$ over R_q. Applying the NTT transform, this equation implies that $\hat{\mathbf{s}}^* \in \{0, 1, 2\}^d$. Finally, by subtracting the second verification equation from one another, we get $\mathbf{A}(\hat{\mathbf{z}}_0^{(1)} - \hat{\mathbf{z}}_0^{(2)}) = (c^{(1)} - c^{(2)}) \cdot \mathbf{u}$. Since $c^{(1)} \neq c^{(2)}$ and we established $z_0^{(k)} = y^* + c^{(k)} \cdot s^*$ for each $k \in [3]$, this implies $\mathbf{A}\hat{\mathbf{s}}^* = \mathbf{u}$ as desired.

To summarize, with probability $1/3$, L contains $((\beta^{(k,j)}, \mathbf{z}^{(k,j)}))_{(k,j)\in[3]\times[2]}$. Conditioned on this fact, $\mathsf{SL\text{-}Extract}$ outputs a valid witness $s^* \in \mathcal{R}_{\mathsf{ES}}$ unless it finds a solution to the $\mathsf{MSIS}_{n,6n,8B_{\mathbf{z}}}$ problem. Note that $\mathsf{SL\text{-}Extract}$ performs all the steps without explicitly knowing $(c^{(k)}, (z_0^{(k)}, x_0^{(k)}, x_1^{(k)}, x_2^{(k)}))_{k\in[3]}$. □

Remark 5.4 (Amplifying soundness). To achieve negligible soundness error, we need to repeat the protocol t-times so that $(3/q+2/d)^t$ is negligible. However, Bootle et al. [BLS19] observed that we can do better. Since the modulus size q is typically much larger than the dimension size d, we can perform t'-parallel execution of the lower half of the protocol for each t-parallel execution of the upper half, resulting in a soundness error of $(3/q + 2/d^{t'})^t$. By taking t' to be $q \approx d^{t'}$, this results in a better soundness amplification.

A keen reader may notice that our protocol in Section 4.2 is not amenable to this technique. Unlike Bootle et al.'s protocol, we "commit" to both vectors $\mathbf{e}$ and $\mathbf{r}$ before obtaining the first challenge $c \in \mathbb{Z}_q$ using the extractable LinHC protocol, where in Bootle et al's protocol, $\mathbf{e}$ and $\mathbf{r}$ were sampled before and after c was given, respectively. Although we cannot apply Bootle et al's technique for general extractable LinHC protocols, we observe that we can apply it when instantiating the extractable LinHC protocol with our two concrete constructions in Section 3. Informally, the property we require to employ this technique is that $\mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}, \mathbf{r}))$ can be decomposed into two independent algorithms: $\mathsf{Com}_{\mathbf{e}}(\mathsf{K}_{\mathsf{com}}, \mathbf{e})$ and $\mathsf{Com}_{\mathbf{r}}(\mathsf{K}_{\mathsf{com}}, \mathbf{r})$, while maintaining zero-knowledge and straight-line extractability even reusing the $\mathsf{Com}_{\mathbf{e}}(\mathsf{K}_{\mathsf{com}}, \mathbf{e})$ on multiple $\mathsf{Com}_{\mathbf{r}}(\mathsf{K}_{\mathsf{com}}, \mathbf{r})$. This property is naturally satisfied by our two instantiations and we provide a concrete example of this in the following section.

5.2 QROM Secure Exact Sound NIZK via Extractable LinHC and Fiat-Shamir

Bootle et al. [BLS19] showed how to transform their interactive protocol into an NIZK using the soundness amplification technique explained in Remark 5.4. We can apply the same technique when instantiating the extractable LinHC protocol with our two constructions provided in Sections 3.4 and 3.5. In this section, as a concrete example, we provide the full detail of our straight-line extractable NIZK by instantiating the extractable LinHC protocol with our second construction in Section 3.5.

Parameters. Parameters $d, q, B_{\mathbf{e}}, B_{\mathbf{r}}$, and $B_{\mathbf{z}}$ are set exactly in the previous section. Let t and t' be positive integers so that $q \approx d^{t'}$ and $(3/q + 2/d^{t'})^t$ is smaller than 2^{-256}. Let $p < q^*$ be coprime odd integers and T be a positive real used by the simplified extractable LinHC protocol in Section 3.5. Following the parameter description provided in Section 3.5, we require $T = \sqrt{12dt}\eta$, $2\sqrt{2}(2pd\eta + (p+1)\sqrt{d})\phi T < q^*/2$, and $\sqrt{2d}\phi T \leq (p-1)/4$.

Our QROM secure exact sound NIZK. For simplicity, we present our NIZK using the simplified extractable LinHC protocol in Section 3.5. Therefore, it only achieves naHVZK and straight-line extractability, independently. Similarly to in Section 4.2. we can use the standard extractable LinHC protocol to make the NIZK simulation straight-line extractable. The prover and verifier algorithms are provided in Figures 13 and 14, respectively. Components with the superscript "*" denote the elements used by the extractable LinHC protocol. Here, we use the standard technique of sending the challenges c_i and β_j instead of the large binding elements $\mathbf{w}_i, \mathbf{w}_j^*, x_{0,j}, x_{1,j}, x_{2,j}$.

Correctness and zero-knowledge of our NIZK follows directly from those of Bootle et al's protocol[25] and the simplified extractable LinHC protocol. We note zero-knowledge of the extractable LinHC protocol when reusing the same $\mathbf{t}_i^*$ can be checked easily from the proof of Lemmata 3.12 and 3.16. Straight-line extractability of our NIZK is a simple generalization of the eu-cma proof of Lemma 4.6 (or more precisely the forgery extraction step, Lemmata 4.7 and 4.8, of the eu-cma proof) combined with the SL-PoK proof in Lemma 5.3. The only difference is that we invoke the GSBP argument in Lemma 2.9 twice to argue that the proof output by the adversary permits the desired structure explained in the beginning of the proof of Lemma 5.3. Namely, we lower bound the probability that the proof leads to 6 valid transcripts in the form explained in Lemma 5.3. The main thing to keep in mind is that once $(\mathbf{t}_i, \mathbf{w}_i, \mathbf{t}_i^*)_{i\in[t]}$ and $(\mathbf{w}_j^*)_{j\in[tt']}$ is fixed, the set of challenges (c, β) for which there exists a valid response is uniquely defined. We omit the full proof since it follows almost exactly the same argument made in the previous proofs.

5.3 Candidate Parameters and Comparison

Asymptotic comparison. We compute the proof size of the NIZK in Figure 13. Each of the t polynomial vectors $\mathbf{t}_i$ (resp. $\mathbf{t}_i^*$) consists of 5 (resp. 6) polynomials in R_q (resp. R_{q^*}). The polynomial $z_{0,i}$ is in R_q. Therefore, the upper half of the proof $(\mathbf{t}_i, \mathbf{t}_i^*, z_{0,i})_{i\in[t]}$ is of size $6td\big(\lceil\log q\rceil + \lceil\log q^*\rceil\big)/8$ bytes. Next, each $\mathbf{z}_j, \mathbf{z}_{j,1}^*, \mathbf{z}_{j,2}^*$ are distributed statistically close to $D^6_{\phi\cdot B_{\mathbf{r}}}, D^6_{\phi\cdot T}, D^6_{\phi\cdot T}$, respectively, so due to Lemma 2.11, these components add up to size $\big(6tt'd(\lceil\log 12\phi B_{\mathbf{r}}\rceil + \lceil\log 12\phi T\rceil)\big)/8$ bytes. Finally, the challenges add up to $(t\lceil\log q\rceil + tt'\lceil\log 2d\rceil)/8$ bytes. Compared to Bootle et al. [BLS19], our proof size is larger by $6td\big(\lceil\log q^*\rceil + t'\lceil\log 12\phi T\rceil\big)/8$ bytes.

Concrete comparison. We provide a comparison between Bootle et al's NIZK and our QROM secure NIZK by considering the application of proving knowledge of the ternary secret in LWE samples over $\mathbb{Z}_q$, which has often been used in the literature to provide a simple benchmark, e.g., [BLS19, Beu20]. Concretely, let $q = 2^{32}$, $d = 2048$ and m be some positive integer smaller than d, and set $\mathbf{A} = [\mathbf{A}'|\mathbf{I}_m] \in \mathbb{Z}_q^{m\times d}$ for a random matrix $\mathbf{A}' \in \mathbb{Z}_q^{m\times(d-m)}$, where these parameters are set to capture the parameter setting of FHE schemes and group signature schemes. We then prove knowledge of a ring element $s \in R_q$ such that its NTT representation $\hat{\mathbf{s}} \in \mathbb{Z}_q^d$ satisfies $\mathbf{A}\hat{\mathbf{s}} = \mathbf{u}$ and $\hat{\mathbf{s}} \in \{0, 1, 2\}^d$. Here, $\hat{\mathbf{s}}$ is understood to be the LWE secret (including the noise). We perform $t' = 3$ lower repetition so that $q \approx d^3 \approx 2^{-31}$ and perform $t = 8$ upper repetitions to reach the 128-bit of post-quantum security level. Then, our NIZK has a proof size of 2071 KB while Bootle et al's has proof size of 812 KB,[26] which is around a factor 2.6 larger. We note that as in [BLS19], we have computed the proof size by using the more space efficient Huffman code to encode the vectors sampled from discrete Gaussian distributions (see [DDLL13] for an example). We provide details on how we set the parameters in Appendix B.2.

Applying the Unruh transform. The standard Unruh transform only works for Σ-protocols but Chen et al. [CHR+18] extended the Unruh transform to work against a 5-round public-coin HVZK interactive protocol when restricting the second challenge to be *binary*. Although we did not check in detail if the extended Unruh transform can be securely applied to Bootle et al's protocol, we computed the proof size assuming it is (see Footnote 4). Specifically, if we were to make Bootle et al's NIZK secure in the QROM using the extended

[25] We found a slight issue with the zero-knowledge proof of Bootle et al's NIZK so the protocol in Figure 13 is modified accordingly. Discussion on the modification is provided in Appendix B.1 for completeness.

[26] Bootle et al. [BLS19] provides a proof size of 384 KB. Ours is around two times larger since we require $t = 8$, unlike $t = 4$, to achieve post-quantum security. Moreover, we do not reuse the commitment $t_{3,i}$ for all $i \in [t]$ as in [BLS19] since it would harm zero-knowledge (see Appendix B.1).

$\mathsf{NIZK.Prove}^{\mathsf{H}}(\mathsf{crs} = (\mathbf{B}, \mathsf{K_{com}}), \mathsf{X} = (\mathbf{A}, \mathbf{u}), \mathsf{W} = s)$

1: **for** $i \in [t]$ **do**
2: $\quad y_i \leftarrow R_q$
3: $\quad \mathbf{w}_i \leftarrow \mathbf{A}\hat{\mathbf{y}}_i \in \mathbb{Z}_q^m$
4: $\quad \mathbf{e}_i = (e_{1,i}, e_{2,i}, e_{3,1}, e_{4,i}, e_{5,i}, e_{6,i})^\top \leftarrow S^6_{B_\mathbf{e}}$
5: $\quad t_{1,i} \leftarrow \mathbf{b}_1^\top \mathbf{e}_i$
6: $\quad t_{2,i} \leftarrow \mathbf{b}_2^\top \mathbf{e}_i + y_i$
7: $\quad t_{3,i} \leftarrow \mathbf{b}_3^\top \mathbf{e}_i + s$
8: $\quad t_{4,i} \leftarrow \mathbf{b}_4^\top \mathbf{e}_i + y_i(2s-3)$
9: $\quad t_{5,i} \leftarrow \mathbf{b}_4^\top \mathbf{e}_i + y_i^2(s-3)$
10: $\quad \mathbf{t}_i \leftarrow (t_{1,i}, t_{2,i}, t_{3,i}, t_{4,i}, t_{5,i})^\top$
11: $\quad$ **for** $\ell \in [6]$ **do**
12: $\quad\quad (s^*_{\ell,i,1}, s^*_{\ell,i,2}) \leftarrow S^2_\eta$
13: $\quad\quad t^*_{\ell,i} \leftarrow h^* s^*_{\ell,i,1} + p \cdot s^*_{\ell,i,2} + e_{\ell,i}$
14: $\quad$ **for** $\omega \in [2]$ **do**
15: $\quad\quad \mathbf{s}^*_{i,\omega} \leftarrow (s^*_{1,i,\omega}, s^*_{2,i,\omega}, s^*_{3,\omega}, s^*_{4,i,\omega}, s^*_{5,i,\omega}, s^*_{6,\omega})^\top$
16: $\quad \mathbf{t}^*_i \leftarrow (t^*_{1,i}, t^*_{2,i}, t^*_{3,i}, t^*_{4,i}, t^*_{5,i}, t^*_{6,i})^\top$
17: **for** $j \in [tt']$ **do**
18: $\quad \mathbf{r}_j \leftarrow D^6_{\phi \cdot B_\mathbf{r}}$
19: $\quad (\mathbf{y}^*_{j,1}, \mathbf{y}^*_{j,2}) \leftarrow D^6_{\phi \cdot T} \times D^6_{\phi \cdot T}$
20: $\quad \mathbf{w}^*_j \leftarrow h^* \cdot \mathbf{y}^*_{j,1} + p \cdot \mathbf{y}^*_{j,2} + \mathbf{r}_j$
21: $(c_i)_{i\in[t]} \leftarrow \mathsf{H}\big(0 \| (\mathbf{t}_i, \mathbf{w}_i, \mathbf{t}^*_i)_{i\in[t]}, (\mathbf{w}^*_j)_{j\in[tt']}\big)$
22: **for** $i \in [t]$ **do**
23: $\quad z_{0,i} \leftarrow c_i \cdot s + y_i$
24: $\quad$ **for** $j \in \{(i-1)t'+1, \cdots, (i-1)t'+t'\}$ **do**
25: $\quad\quad x_{0,j} \leftarrow \mathbf{b}_1^\top \mathbf{r}_j$
26: $\quad\quad x_{1,j} \leftarrow (\mathbf{b}_2^\top + c_i \mathbf{b}_3^\top)\mathbf{r}_j$
27: $\quad\quad x_{2,j} \leftarrow \big((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot \mathbf{b}_3^\top$
28: $\quad\quad\quad\quad - z_{0,j} \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top\big)\mathbf{r}_j$
29: $(\beta_j)_{j\in[tt']} \leftarrow \mathsf{H}\big(1 \| ((\mathbf{t}_i, \mathbf{w}_i, \mathbf{t}^*_i)_{i\in[t]}, (\mathbf{w}^*_j)_{j\in[tt']}), (c_i)_{i\in[t]},$
30: $\quad\quad\quad\quad \big((z_{0,i})_{i\in[t]}, (x_{0,j}, x_{1,j}, x_{2,j})_{j\in[tt']}\big)\big)$
31: **for** $j \in [tt']$ **do**
32: $\quad \mathbf{z}_j \leftarrow \beta_j \cdot \mathbf{e}_{\lceil j/t' \rceil} + \mathbf{r}_j$
33: $\quad \mathbf{z}^*_{j,1} \leftarrow \beta_j \cdot \mathbf{s}_{\lceil j/t' \rceil,1} + \mathbf{y}^*_{j,1}$
34: $\quad \mathbf{z}^*_{j,2} \leftarrow \beta_j \cdot \mathbf{s}_{\lceil j/t' \rceil,2} + \mathbf{y}^*_{j,2}$
35: **for** $k \in [t']$ **do**
36: $\quad \bar{\mathbf{z}}_k \leftarrow [\mathbf{z}_k \| \mathbf{z}_{t'+k} \| \cdots \| \mathbf{z}_{(t-1)t'+k}]$
37: $\quad \bar{\mathbf{e}}_k \leftarrow [\beta_k \mathbf{e}_1 \| \beta_{t'+k} \mathbf{e}_2 \| \cdots \| \beta_{(t-1)t'+k} \mathbf{e}_t]$
38: $\quad b \leftarrow \mathsf{Rej}(\bar{\mathbf{z}}_k, \bar{\mathbf{e}}_k, \phi, B_\mathbf{r}, \mathsf{err})$
39: $\quad$ **for** $\omega \in [2]$ **do**
40: $\quad\quad \bar{\mathbf{z}}^*_{k,\omega} \leftarrow [\mathbf{z}^*_{k,\omega} \| \mathbf{z}^*_{t'+k,\omega} \| \cdots \| \mathbf{z}^*_{(t-1)t'+k,\omega}]$
41: $\quad\quad \bar{\mathbf{s}}^*_{k,\omega} \leftarrow [\beta_k \mathbf{s}^*_{1,\omega} \| \beta_{t'+k} \mathbf{s}^*_{2,\omega} \| \cdots \| \beta_{(t-1)t'+k} \mathbf{s}^*_{t,\omega}]$
42: $\quad b^* \leftarrow \mathsf{Rej}([\bar{\mathbf{z}}^*_{k,1} \| \bar{\mathbf{z}}^*_{k,2}], [\bar{\mathbf{s}}^*_{k,1} \| \bar{\mathbf{s}}^*_{k,2}], \phi, T, \mathsf{err})$
43: $\quad$ **if** $b = \perp \ \vee \ b^* = \perp$ **then**
44: $\quad\quad$ **goto** Line 1
45: **return** $\pi := \big((\mathbf{t}_i, \mathbf{t}^*_i, c_i, z_{0,i})_{i\in[t]},$
46: $\quad\quad\quad\quad (\beta_j, \mathbf{z}_j, \mathbf{z}^*_{j,1}, \mathbf{z}^*_{j,2})_{j\in[tt']}\big)$

Figure 13: Prover algorithm of the QROM secure exact sound NIZK for the relation $\mathcal{R}_{\mathsf{ES}}$ in the CRS model. The statement $\mathsf{X} = (\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{m \times d} \times \mathbb{Z}_q^m$ and witness $s \in R_q$ satisfy $\mathbf{A}\hat{\mathbf{s}} = \mathbf{u}$ and $\hat{\mathbf{s}} \in \{0,1,2\}^d$. $\mathsf{K_{com}} = h^* \in R_{q^*}$ is the commitment key of the simplified extractable LinHC protocol in Section 3.5, $\mathbf{B} \in R_q^{5\times 6}$ is the public parameter of the (implicit) commitment scheme Π_{Com} (see Appendix A.4), and $\mathbf{b}_i^\top$ and $b_{i,j}$ denotes its i-th row vector and (i,j)-th entry, respectively. The gray indicates the components that are used in the protocol of Bootle et al. [BLS19].

Unruh transform, the proof size would be 44.9MB, which is around a factor 51.8 larger compared to Bootle et al's NIZK secure in the classical ROM. For completeness, we provide the details in Appendix C.3. Finally, note that it is unclear whether the Fiat-Shamir transform in the QROM can be securely applied to Bootle et al's NIZK.

5.4 Further Applications of Extractable LinHC

We show that other recent Σ-/public-coin HVZK interactive protocols are compatible with our extractable LinHC protocol. As the main focus of this work is introducing the concept of extractable LinHC protocols and providing another route to obtaining QROM security, we leave optimization and assessment of the concrete security of these other protocols as future work.

[BDL+18]: Opening to commitments. The commitment scheme by Baum et al. is used in almost all recent lattice-based ZK proofs (See Appendix A.4). Since this was also implicitly used in [BLS19], it is clear that we can turn the Σ-protocol of valid opening of a commitment to a QROM secure NIZK.

[ESLL19]: Range proofs. Range proof allows one to prove that a committed value resides in a specific range and is used in applications such as privacy-preserving linkable anonymous credentials and confidential transactions in cryptocurrencies. Recently, Esgin et al. [ESLL19] provided an efficient range proof by using new ideas on CRT-packing supporting "inter-slot" operations and NTT-friendly tools that permit the use

$\mathsf{NIZK.Verify}^{\mathsf{H}}(\mathsf{crs}, \mathsf{X}, \pi)$

$\left((\mathbf{t}_i, \mathbf{t}_i^*, c_i, z_{0,i})_{i\in[t]}, (\beta_j, \mathbf{z}_j, \mathbf{z}_{j,1}^*, \mathbf{z}_{j,2}^*)_{j\in[tt']}\right) \leftarrow \pi$
for $i \in [t]$ **do**
  $\mathbf{w}_i \leftarrow \mathbf{A}\hat{\mathbf{z}}_{0,i} - c_i \cdot \mathbf{u}$
  for $j \in \{(i-1)t'+1, \cdots, (i-1)t'+t'\}$ **do**
    $\mathbf{w}_j^* \leftarrow h^* \cdot \mathbf{z}_{j,1}^* + p \cdot \mathbf{z}_{j,2}^* - \beta_j \cdot \mathbf{t}_i^* + \mathbf{z}_j$
    $x_{0,j} \leftarrow \mathbf{b}_1^\top \mathbf{z}_i - \beta_j \cdot t_{1,i}$
    $x_{1,j} \leftarrow (\mathbf{b}_2^\top + c_i \mathbf{b}_3^\top)\mathbf{z}_j - \beta_j \cdot (c_i \cdot t_{3,i} + t_{2,i})$
    $x_{2,j} \leftarrow \left((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot \mathbf{b}_3^\top - z \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top\right)\mathbf{z}_j - \beta_j \cdot ((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot t_{3,i} - z_{0,i} \cdot t_{4,i} + t_{5,i})$
$(c_i')_{i\in[t]} \leftarrow \mathsf{H}\big(0 \| (\mathbf{t}_i, \mathbf{w}_i)_{i\in[t]}, (\mathbf{w}_j^*)_{j\in[tt']}\big)$
$(\beta_j')_{j\in[tt']} \leftarrow \mathsf{H}\big(1 \| ((\mathbf{t}_i, \mathbf{w}_i, \mathbf{t}_i^*)_{i\in[t]}, (\mathbf{w}_j^*)_{j\in[tt']}), (c_i')_{i\in[t]}, \big((z_{0,i})_{i\in[t]}, (x_{0,j}, x_{1,j}, x_{2,j})_{j\in[tt']}\big)\big)$
for $j \in [tt']$ **do**
  if $\|\mathbf{z}_{j,1}^*\|_2 > \sqrt{12d} \cdot \phi \cdot T \ \vee \ \|\mathbf{z}_{j,2}^*\|_2 > \sqrt{12d} \cdot \phi \cdot T$ **then**
    return $\bot$
  if $\|\mathbf{z}_j\|_2 > B_{\mathbf{z}}$ **then**
    return $\bot$
if $(c_i)_{i\in[t]} \neq (c_i')_{i\in[t]} \ \vee \ (\beta_j)_{j\in[tt']} \neq (\beta_j')_{j\in[tt']}$ **then**
  return $\bot$
else
  return $\top$

Figure 14: Verifier algorithm of the QROM secure exact sound NIZK for the relation $\mathcal{R}_{\mathsf{ES}}$ in the CRS model. The gray indicates the components that are used in the protocol of Bootle et al. [BLS19].

of fully-splitting rings. It can be checked that the Σ-protocol for the range relation provided in [ESLL19, Theorem 1] is compatible with extractable LinHC protocols. Although it was not necessary for their scheme, we can modify the verifier in [ESLL19, Protocol 2] (without affecting any parameters) to further check the bound on $\boldsymbol{f}_{\mathrm{crt}}$ to perfectly fit the description of the extractable LinHC protocol. Concretely, we can view $(a_j^i)_{(i,j)\in[\psi,k_i-1]}, \boldsymbol{r}_a, \boldsymbol{r}_d$, and $\boldsymbol{r}_e$ in their Protocol 2 as $\mathbf{r}$, and $(b_j^i)_{(i,j)\in[\psi,k_i-1]}, \boldsymbol{r}_b, \boldsymbol{r}_c$, and $\boldsymbol{r}$ in their Protocol 2 as $\mathbf{e}$ of the extractable LinHC protocol in our Figure 3.

[ESLL19]: One-out-of-many proofs. One-out-of-many proofs is a ubiquitous tool that allows to construct many advanced signature schemes such as group signatures and ring signatures. In the same paper as above, Esgin et al. [ESLL19] also provided an efficient one-out-of-many proofs building on similar ideas. They first construct a Σ-protocol to prove that a given commitment opens to a binary string and then use it as a building block to construct a Σ-protocol for the one-out-of-many proof relation. Similarly to above, it can be checked that both corresponding Σ-protocols provided in [ESLL19, Theorems 2 and 3] are compatible with extractable LinHC protocols. We note this is the only protocol that we are aware of that has a response of the form $\mathbf{z} = \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}$. All other protocols have the form $\mathbf{z} = \beta \cdot \mathbf{e} + \mathbf{r}$.

[YAZ+19]: Exact sound proofs for quadratic relations. In an independent and concurrent work to [BLS19], Yang et al. [YAZ+19] provided an exact sound proof for the relation $\mathcal{R}_{\mathsf{QES}}$, such that $\mathsf{X} = (\mathbf{A}, \mathbf{u}, \mathcal{M}) \in \mathbb{Z}_q^{m\times d} \times \mathbb{Z}_q^m \times ([1,d]^3)^\ell$ and $\mathsf{W} = \mathbf{s} \in \mathbb{Z}_q^n$ satisfies $(\mathsf{X}, \mathsf{W}) \in \mathcal{R}_{\mathsf{QES}}$ if and only if $\mathbf{As} = \mathbf{u}$ and $s_h = s_i \cdot s_j$ for $(h,i,j) \in \mathcal{M}$. Here, $\mathcal{M}$ is a set of ℓ triples that defines quadratic constraints over $\mathbf{s}$ and s_i denotes the i-th entry of $\mathbf{s}$. They then showed that many useful relations such as possession of short secret can be embedded into such a relation. It can be checked that the Σ-protocol for the relation $\mathcal{R}_{\mathsf{QES}}$ in [YAZ+19, Figure 2] can be easily modified to be compatible with extractable LinHC protocols, similarly to [BLS19]. Note that Yang et al. commits the first message sent by the prover by a hash function modeled as a random oracle to achieve standard HVZK rather than naHVZK. However, since naHVZK is sufficient for constructing NIZK and signatures, we can rewrite the Σ-protocol of Yang et al. so that the prover simply sends the commitment $\boldsymbol{t}, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$, and $\boldsymbol{c}_4$ as the first message. Then, we can follow a simplified argument we made in Sections 5.1 and 5.2 to prove quantum security of their protocol.

[ALS20]: Product proofs for commitments. Being able to prove relationships between committed messages is very useful. For instance, if we can prove that committed messages satisfy additive and multiplicative relations, then we can prove satisfiability of boolean/arithmetic circuits in zero-knowledge, which can then in turn be used to construct advanced signature schemes such as attribute-based signatures. Recently, Attema et al. [ALS20] provided an efficient Σ-/interactive protocol for proving multiplicative relations for commitments. We observe that the two Σ-protocols provided in [ALS20, Figures 2 and 3] perfectly fit the description of the extractable LinHC protocol. We suspect the most general 5-round interactive protocol provided in [ALS20, Figures 4] can be made quantum secure. However, since the scheme employs several complex optimizations, it is not clear if it is compatible with our current formalization of extractable LinHC protocols.

Finally, we like to elucidate one notable aspect of the recent advanced lattice-based protocols. While conventional Σ-protocols only require 2 to 3 valid transcripts to invoke special soundness, we require as much as 32 valid transcripts in the recent protocols, e.g., [ALS20]. Therefore, even if were able to show that the Σ-protocol had a compatible lossy function as in the definition of [LZ19], the Fiat-Shamir transform incurs an extremely large reduction loss. Concretely, combining [DFMS19, Lemma 29] and [LZ19, Theorem 1], a knowledge extractor (for the Σ-protocol) that is given oracle access to a quantum adversary that outputs a valid NIZK proof with probability ϵ after making Q oracle queries, is only guaranteed to succeed in extracting a witness with probability $(\epsilon/Q^2)^{2\times 32-1} = \epsilon^{63}/Q^{126}$. In such cases, extractable LinHC protocols can provide a much tighter proof and a smaller set of provably secure parameters.

Acknowledgement. Shuichi Katsumata was supported by JST CREST Grant Number JPMJCR19F6. We thank Thomas Prest, Alexandre Wallet, and Thomas Espitau for helpful inputs on NTRU. We also want to thank Patrick Hough for helpful discussions about this work while he visited AIST in 2020.

References

[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. 3, 4, 5, 8, 9

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 669–699. Springer, Heidelberg, August 2018. 3, 4, 9

[BCC04] Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004*, pages 132–145. ACM Press, October 2004. 3

[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.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. 3, 7, 12

[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. 3, 4, 5, 8

[Bel20] Mihir Bellare. Lectures on nizks: A concrete security treatment. Lecture Notes, 2020. Available at https://cseweb.ucsd.edu/ mihir/cse208-Wi20/main.pdf. 10

[Beu20] Ward Beullens. Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 183–211. Springer, Heidelberg, May 2020. 9

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 441–469. Springer, Heidelberg, August 2020. 9

[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. 2, 3, 4, 5, 8, 9

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, Heidelberg, May 2003. 3

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012. 13

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013. 3

[CC18] Pyrros Chaidos and Geoffroy Couteau. Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 193–221. Springer, Heidelberg, April / May 2018. 9

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1825–1842. ACM Press, October / November 2017. 3, 9

[CHR+16] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass MQ-based identification to MQ-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 135–165. Springer, Heidelberg, December 2016. 9

[CHR+18] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. SOFIA: $\mathcal{MQ}$-based signatures in the QROM. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 3–33. Springer, Heidelberg, March 2018. 4, 8, 9

[CPSV16] Michele Ciampi, Giuseppe Persiano, Luisa Siniscalchi, and Ivan Visconti. A transform for NIZK almost as efficient and general as the Fiat-Shamir transform without programmable random oracles. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 83–111. Springer, Heidelberg, January 2016. 9

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56. Springer, Heidelberg, August 2013.

[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. 3, 9, 29

[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. 3, 4, 7, 8, 9, 29, 30

[DFN06] Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 41–59. Springer, Heidelberg, March 2006. 9

[EKP20] Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 157–186. Springer, Heidelberg, May 2020. 9

[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. 9

[ESLL19] Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 115–146. Springer, Heidelberg, August 2019. 3, 4, 5, 8, 16

[ESS+19] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 67–88. Springer, Heidelberg, June 2019.

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Heidelberg, August 2005. 5, 6, 18

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 60–79. Springer, Heidelberg, December 2012. 4, 15

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. 3, 4

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. 13

[GKZ19] Alex B Grilo, Iordanis Kerenidis, and Timo Zijlstra. Learning-with-errors problem is easy with quantum samples. *Physical Review A*, 99(3):032314, 2019. 14

[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. 8

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996. 24

[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. 14

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Heidelberg, March 2016. 8

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007. 9

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 525–537. ACM Press, October 2018. 9

[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. 3, 4, 7, 8, 9, 13, 30

[KM07] Neal Koblitz and Alfred J. Menezes. Another look at "provable security". *Journal of Cryptology*, 20(1):3–37, January 2007. 3

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, Heidelberg, December 2008. 4

[KZ20] Daniel Kales and Greg Zaverucha. An attack on some signature schemes constructed from five-pass identification schemes. In Stephan Krenn, Haya Shulman, and Serge Vaudenay, editors, *CANS 20*, volume 12579 of *LNCS*, pages 3–22. Springer, Heidelberg, December 2020. 3, 9

[Lin15] Yehuda Lindell. An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 93–109. Springer, Heidelberg, March 2015. 9

[LLM+16] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 101–131. Springer, Heidelberg, December 2016. 9

[LLNW16] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 1–31. Springer, Heidelberg, May 2016. 9

[LLNW17] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based PRFs and applications to E-cash. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 304–335. Springer, Heidelberg, December 2017. 9

[LN17] Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 293–323. Springer, Heidelberg, April / May 2017. 9

[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 20*, pages 1051–1070. ACM Press, November 2020. 9

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, Heidelberg, February / March 2013. 9

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015. 14

[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. 5, 14

[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. 3, 4, 5, 8, 27

[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. 3, 4, 5, 8, 13, 14, 27

[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. 3, 7, 8, 9, 29, 30

[Mau15] Ueli Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *Designs, Codes and Cryptography*, 77(2):663–676, 2015. 9

[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. 14

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 12

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 8

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009. 14

[Ste94] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, Heidelberg, August 1994. 4

[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. 5, 13, 14

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012. 3

[Unr15] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 755–784. Springer, Heidelberg, April 2015. 3, 4, 5, 6, 7

[Unr17] Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 65–95. Springer, Heidelberg, December 2017. 3, 7

[YAZ+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 147–175. Springer, Heidelberg, August 2019. 3, 4, 5, 8

[Zha12a] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012. 8, 13

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012. 3, 31

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019. 3, 12

A Omitted Preliminary

In this section, we provide the omitted preliminaries.

A.1 The MSIS Assumption

The (module) short integer solution problem is defined as follows.

Definition A.1 (MSIS). *For integers $n = n(\kappa), m = m(n), q = q(n) > 2$, a positive real B, and a QPT algorithm $\mathcal{A}$, the advantage of the module short integer solution problem $\mathsf{MSIS}_{n,m,B}$ of $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}^{\mathsf{MSIS}_{n,m,B}}(\mathcal{A}) = \left|\Pr[\mathbf{A} \leftarrow R_q^{n\times m}, \mathbf{u} \leftarrow \mathcal{A}(\mathbf{A}) : \mathbf{Au} = \mathbf{0} \ \wedge \ 0 < \|\mathbf{u}\|_2 \leq B]\right|.$$

A.2 Classical Lyubashevsky's Σ-Protocol for a Basic Lattice Relation

We provide Lyubashevsky's original Σ-protocol for the relations $(\mathcal{R}_{\mathsf{MSIS}}, \mathcal{R}'_{\mathsf{MSIS}})$ such that $\mathcal{R}_{\mathsf{MSIS}} \subseteq \mathcal{R}'_{\mathsf{MSIS}}$ [Lyu09, Lyu12] in Figure 15.

It is known that this Σ-protocol satisfies naHVZK and relaxed 2-special soundness. Roughly these two security properties are argued as follows: For naHVZK, assume the zero-knowledge simulator ZKSim is given a random challenge $\beta \in \mathsf{ChSet}$. It then samples $\mathbf{z} \leftarrow D^m_{\phi \cdot B_{\mathbf{r}}}$ and returns $(\mathbf{w} = \mathbf{Az} - \beta \cdot \mathbf{u}, \mathbf{z})$. Due to rejection sampling, this is statistically indistinguishable from the real transcript conditioned on not aborting. Next, for special soundness, assume we are given $(\mathbf{w}, \beta, \widetilde{\beta}, \mathbf{z}, \widetilde{\mathbf{z}})$ for $\beta \neq \widetilde{\beta}$. Then the special soundness extractor $\mathsf{Extract}_{\mathsf{ss}}$ outputs the witness $\mathbf{z}^* = \mathbf{z} - \widetilde{\mathbf{z}}$ with approximation factor $\beta^* = \beta - \widetilde{\beta}$. It is clear that $\mathbf{Az}^* = \beta^* \cdot \mathbf{u}$, $\|\mathbf{z}^*\|_2 \leq 2 \cdot B_{\mathbf{z}}$. This establishes that $((\mathbf{A}, \mathbf{u}), \mathbf{z}^*) \in \mathcal{R}'_{\mathsf{MSIS}}$.

A.3 Background on Signature Scheme

In this work, we consider deterministic signature schemes; schemes where the singing algorithm is deterministic. This is without loss of generality since any randomized signing algorithm can be derandomized by deriving message-specific randomness by a PRF.

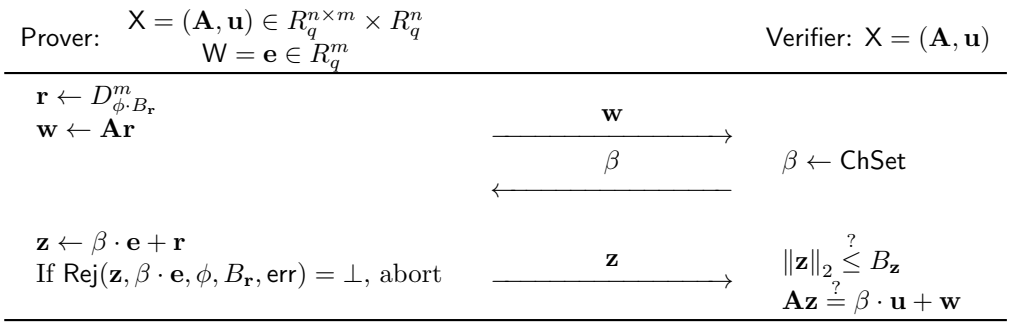

Figure 15: Lyubashevsky's Σ-protocol for the lattice relation $\mathbf{A}\mathbf{e} = \mathbf{u}$. The witness $\mathbf{e}$ satisfies $\mathbf{e}$ $\|\mathbf{e}\|_2 \leq B_\mathbf{e}$. In case abort occurs, we send $\perp$ as the third message.

Definition A.2 (Signature scheme in the QROM). *A signature scheme Π_{Sig} in the QROM with message space $\mathcal{M}$ is a tuple of PPT algorithms $\Pi_{\mathsf{Sig}} = (\mathsf{S.KeyGen}, \mathsf{S.Sign}, \mathsf{S.Verify})$ with oracle access to a random oracle H defined as follows:*

$\mathsf{S.KeyGen}^{\mathsf{H}}(1^\kappa) \to (\mathsf{vk}, \mathsf{sk})$: *The key generation algorithm takes as input the security parameter 1^κ and outputs a verification key vk and signing key sk.*

$\mathsf{S.Sign}^{\mathsf{H}}(\mathsf{vk}, \mathsf{sk}, \mathsf{M}) \to \sigma$: *The* deterministic *signing algorithm takes as inputs the verification key vk, signing key sk and message $\mathsf{M} \in \mathcal{M}$, and outputs a signature σ.*

$\mathsf{S.Verify}^{\mathsf{H}}(\mathsf{vk}, \mathsf{M}, \sigma) \to \top$ **or** $\perp$: *The* deterministic *verification algorithm takes as inputs the verification key vk, message $\mathsf{M} \in \mathcal{M}$ and signature σ, and outputs $\top$ if the signature is valid and outputs $\perp$ otherwise.*

Correctness. We say a signature scheme has correctness error δ if for all $\kappa \in \mathbb{N}$, messages $\mathsf{M} \in \mathcal{M}$, we have $\Pr[\mathsf{S.Verify}^{\mathsf{H}}(\mathsf{vk}, \mathsf{M}, \sigma) \neq \perp] \geq 1 - \delta$, where the probability is taken over the randomness to sample H, $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}^{\mathsf{H}}(1^\kappa)$, and $\sigma \leftarrow \mathsf{S.Sign}(\mathsf{sk}, \mathsf{M})$.

Security. We define the standard existential unforgeability under a chosen message attack ($\mathsf{eu\text{-}cma}$) security. The security notion is defined by the following game between an adversary $\mathcal{A}$ and a challenger. We assume a random oracle H is randomly chosen prior to the game and the challenger and adversary are granted quantum access to it.

Setup: The challenger runs $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{S.KeyGen}^{\mathsf{H}}(1^\kappa)$ and gives vk to $\mathcal{A}$. The challenger also initializes an empty set S^{msg}.

Signature Query: When $\mathcal{A}$ submits a message $\mathsf{M} \in \mathcal{M}$, the challenger runs $\sigma \leftarrow \mathsf{S.Sign}^{\mathsf{H}}(\mathsf{vk}, \mathsf{sk}, \mathsf{M})$ and returns the signature σ to $\mathcal{A}$. It further updates $S^{\mathsf{msg}} = S^{\mathsf{msg}} \cup \{\mathsf{M}\}$.

Output: Finally, $\mathcal{A}$ outputs a pair (M^*, σ^*). The adversary $\mathcal{A}$ wins if the following conditions are met:

- $\mathsf{M}^* \notin S^{\mathsf{msg}}$,
- $\mathsf{S.Verify}^{\mathsf{H}}(\mathsf{vk}, \mathsf{M}^*, \sigma^*) = \top$.

We say the signature scheme Π_{Sig} is $\mathsf{eu\text{-}cma}$ secure if the advantage $\mathsf{Adv}^{\mathsf{eu\text{-}cma}}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}]$ is negligible for any QPT $\mathcal{A}$.

A.4 Background on Commitment Scheme

Although the usage of the commitment scheme of Baum et al. [BDL+18] is only implicit in Section 5, we provide the detail for completeness.

Definition A.3 (Commitment scheme). *A commitment scheme is a tuple of PPT algorithms* $\Pi_{\mathsf{Com}} = (\mathsf{C.Setup}, \mathsf{C.Com}, \mathsf{C.Open})$ *defined as follows:*

$\mathsf{C.Setup}(1^\kappa) \to \mathsf{pp}$: *The setup algorithm takes as input the security parameter* 1^κ *and outputs public parameter* pp.

$\mathsf{C.Com}(\mathsf{pp}, \mathsf{M}) \to (\mathsf{com}, \mathsf{open})$: *The commitment algorithm takes as input the public parameter* pp *and message* M*, and outputs a commitment* com *and an opening* open.

$\mathsf{C.Open}(\mathsf{pp}, \mathsf{M}, \mathsf{com}, \mathsf{open}) \to \top$ **or** $\bot$**:** *The deterministic opening algorithm takes as input the public parameter* pp*, message* M*, commitment* com *and opening* open*, and outputs* $\top$ *if* open *is a valid opening and outputs* $\bot$ *otherwise.*

Hiding. We say a commitment scheme is ϵ_{hide}-*hiding* if for all QPT algorithms $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{hide}}(\mathcal{A})$ defined below is less than ϵ_{hide}:

$$\mathsf{Adv}^{\mathsf{hide}}(\mathcal{A}) := \left| \Pr\left[b = b' : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{C.Setup}(1^\kappa), (\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A}(\mathsf{pp}) \\ b \leftarrow \{0,1\}, (\mathsf{com}, \mathsf{open}) \leftarrow \mathsf{C.Com}(\mathsf{pp}, \mathsf{M}_b) \\ b' \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{com}) \end{array} \right] - \frac{1}{2} \right|.$$

Binding. We say a commitment scheme is ϵ_{bind}-*binding* if for all QPT algorithms $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{bind}}(\mathcal{A})$ defined below is less than ϵ_{bind}:

$$\mathsf{Adv}^{\mathsf{bind}}(\mathcal{A}) := \left| \Pr\left[\begin{array}{c} \mathsf{M}_0 \neq \mathsf{M}_1 \ \wedge \ v_0 = v_1 = \top, \text{ where} \\ v_b \leftarrow \mathsf{C.Open}(\mathsf{pp}, \mathsf{M}_b, \mathsf{com}, \mathsf{open}_b) \text{ for } b \in \{0,1\} \end{array} : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{C.Setup}(1^\kappa), \\ (\mathsf{com}, (\mathsf{M}_b, \mathsf{open}_b)_{b\in\{0,1\}}) \leftarrow \mathcal{A}(\mathsf{pp}) \end{array} \right] \right|.$$

We consider a specific commitment scheme used by [BLS19], which in particular is one instantiation of the commitment scheme of Baum et al. [BDL+18]. Their commitment scheme allows to commit to four different ring elements $\mathbf{m} = (m_2, m_3, m_4, m_5)^\top \in R_q^4$. Below, let C denote the set $\{0, X^i \mid 0 \leq i < 2d\} \subset R_q$ and ΔC denote the set $\{a - b \mid a, b \in C\}$. The important property often used in lattice-based cryptography is that, when d is a power of two where $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, for any element in $f \in \Delta C$, $(2f)^{-1}$ exists and has ternary coefficients in $\{-1, 0, 1\}$ [BCK+14]. Finally, let B be some positive real that dictates the hardness of the MSIS problem, whose concrete value is irrelevant right now.

$\mathsf{C.Setup}(1^\kappa)$**:** Sample a random matrix $\mathbf{B} \in R_q^{5\times 6}$ of the following form and output $\mathsf{pp} = \mathbf{B}$:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \\ \mathbf{b}_3^\top \\ \mathbf{b}_4^\top \\ \mathbf{b}_5^\top \end{pmatrix} = \begin{pmatrix} 1 & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} \\ 0 & 1 & 0 & 0 & 0 & b_{2,6} \\ 0 & 0 & 1 & 0 & 0 & b_{3,6} \\ 0 & 0 & 0 & 1 & 0 & b_{4,6} \\ 0 & 0 & 0 & 0 & 1 & b_{5,6} \end{pmatrix}.$$

$\mathsf{C.Com}(\mathsf{pp}, \mathbf{m})$**:** Parse $(m_2, m_3, m_4, m_5)^\top \leftarrow \mathbf{m} \in R_q^4$, sample vector $\mathbf{e} \leftarrow \chi^{6n}$ and compute

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{pmatrix} = \mathbf{B}\mathbf{e} + \begin{pmatrix} 0 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{pmatrix}.$$

Finally, output the commitment $\mathsf{com} = \mathbf{t}$ and opening $\mathsf{open} = (1, \mathbf{e})$. Here, the noise distribution of χ is scheme specific.

$\mathsf{C.Open}(\mathsf{pp}, \mathbf{m}, \mathsf{com}, \mathsf{open})$: Parse $(f, \mathbf{e}) \leftarrow \mathsf{open}$. Check $\|\mathbf{e}\|_2 \leq B$ and $f \in \Delta C$ or $f = 1$ and output $\perp$ if it does not hold. Otherwise, further check $f \cdot \mathbf{t} = \mathbf{B}\mathbf{e} + f \cdot \begin{pmatrix} 0 \\ \mathbf{m} \end{pmatrix}$ and output $\top$ if it holds and $\perp$ otherwise.

In above, the opening algorithm is "relaxed" in some sense since we allow $f \neq 1$ that never occurs when running the commitment algorithm honestly. However, it is known that such an opening algorithm suffices for many situations, and in particular, we have the following.

Lemma A.4 ([BDL+18, Lemmas 6 and 7]). *Assuming the hardness of the* $\mathsf{MLWE}_{n,5n,\chi}$ *and* $\mathsf{MSIS}_{n,6n,8B}$ *problem, the above commitment scheme* Π_{Com} *is hiding and binding, respectively.*

B Omitted Details from Section 5

B.1 Recap: Exact Sound Proof by [BLS19]

We provide a minimal exposition on the work of Bootle et al. [BLS19]. For the full details we refer the readers to the original paper. For completeness, the interactive protocol and the prover and verifier algorithms of the NIZK protocol of Bootle et al. are provided in Figures 16 to 18, respectively. HVZK of the protocol follows from the MLWE assumption and the fact that the underlying (implicit) commitment scheme of Baum et al. [BDL+18] is hiding. The proof of soundness is implicit in our proof of Lemma 5.3.

We note one difference between the NIZK presented in Figure 17 and those presented in [BLS19]. In [BLS19], it was stated that they are able to reuse the same $t_{3,i}$ across all repetition of $i \in [t]$. Looking at the specifics of the underlying commitment scheme in Appendix A.4, this amounts to fixing one $(e_3, e_6) \leftarrow S^2_{B_{\mathbf{e}}}$ once and for all and using the same $t_3 = e_3 + b_{3,6}e_6 + s$ for all $\mathbf{t}_i$. This has the benefit of lowering the proof size by roughly $t\lceil \log q \rceil$ as it no longer needs to recommit to the message s in the t-repetitions.

However, we were not able to prove zero-knowledge of such a scheme so we removed this optimization in Figure 17. The main issue is that if we fix one t_3 once and for all, then due to the specifics of the underlying commitment scheme, e_6 will be reused in all $\{t_{1,i}, t_{2,i}, t_{4,i}, t_{5,i}\}_{i\in[t]}$. Then, for instance, an adversary obtains a set $\{t_{2,i} = e_{2,i} + b_{2,6}e_6 + y_i\}_{i\in[t]}$, where the LWE secret e_6 is reused in t-samples. Although, we are not aware of any practical attacks, the same zero-knowledge proof provided for the interactive case no longer holds unless we assume the LWE problem that allows to reuse the secret. We note that although hardness of such LWE problem can be shown by increasing the noise $e_{2,i}$ by using rerandomization techniques, we will no longer be able to appeal to the ternary secret/noise LWE problem as used in [BLS19].

B.2 Setting the Parameter

One of the benefits of our approach is that we do not have to modify the parameters provided by the underlying interactive proof. Specifically, the extractable LinHC protocol simply works as a wrapper around the underlying non-quantum secure protocol. Therefore, to assess the proof size of our NIZK, we only need to calculate the additional proof size incurred by the extractable LinHC protocol.

First, we recall the parameters used by Bootle et al. [BLS19]: (modulus size) $q = 2^{32}$, (dimension) $d = 2^{11}$, (LWE parameter) $B_{\mathbf{e}} = 1$, (Euclidean norm of $\mathbf{e}_i$) $B_{\mathbf{r}} = 183.83$, (repetition time of upper half and lower half) $(t, t') = (4, 3)$, (bound on Euclidean norm of $\mathbf{z}$ checked by verifier) $B_{\mathbf{z}} = \phi \cdot \sqrt{12 \times 2^{11}} \cdot 183.83$ and (rejection sampling parameter) $\phi = 5$. Here, we note that $\phi = 5$ may to be too small since it results in rejecting the transcript with probability roughly $11/12$ for a *single* repetition. Specifically, the chance of accepting the transcript in $t' = 3$ repetition is only $(1/12)^3$. In our parameter choice, we set a much larger $\phi = 30$ so that the rejection probability of a single repetition is roughly only $1/3$. We note that the parameters given in Section 5.3 is based on the original parameter choice of [BLS19].

We set the remaining parameters required by the extractable LinHC protocol according to the parameter requirement explained in Section 5.2: (repetition time of upper half and lower half) $(t, t') = (8, 3)$, (rejection sampling parameter) $\phi = 30$, (modulus used by LinHC) $(q^*, p) = (2^{50}, 2^{22})$, (MLWE and DSMR parameter for LinHC) $\eta = 1$. Following the same computation provided in Bootle et al. [BLS19, Section 4.1], we evaluate

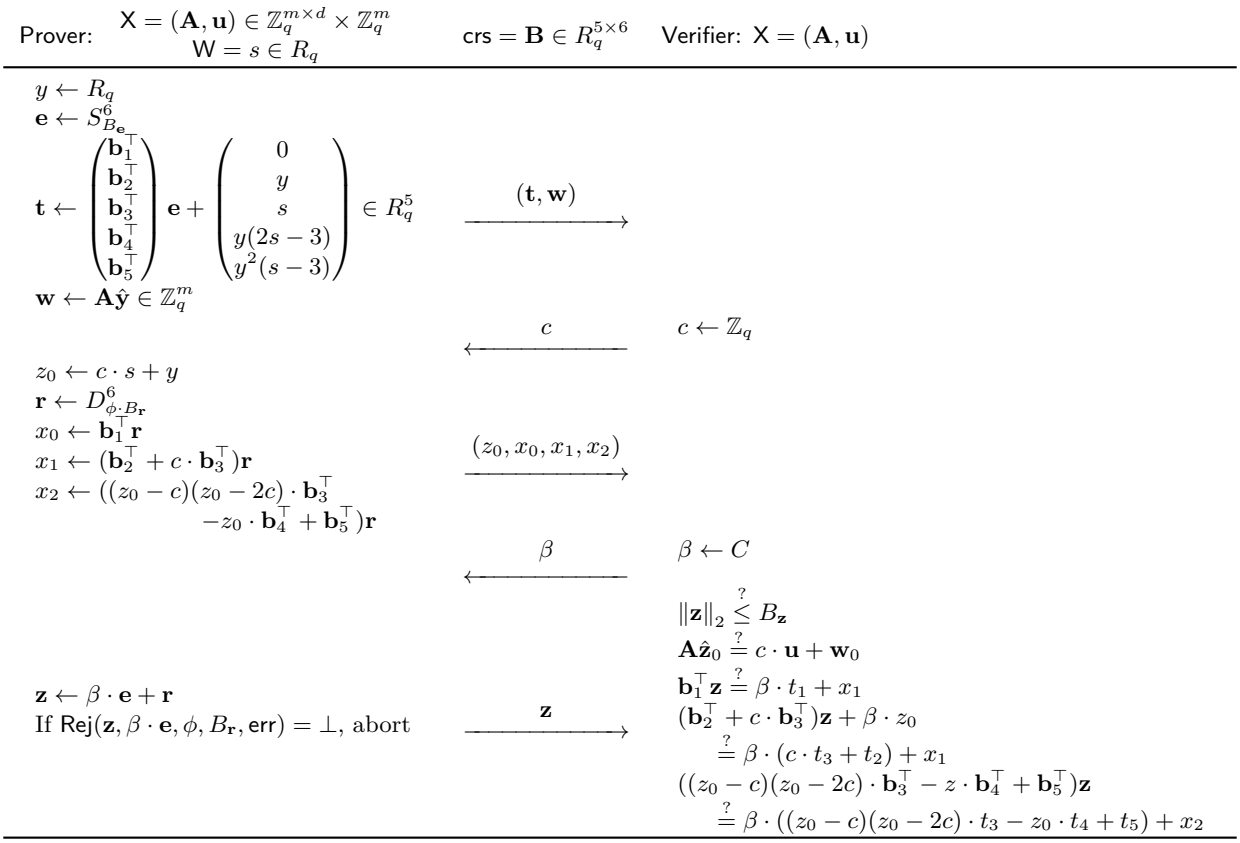

Figure 16: Bootle et al.'s [BLS19] exact sound public-coin interactive protocol in the CRS model. The witness s satisfies $\mathbf{A}\hat{\mathbf{s}} = \mathbf{u}$ and $\hat{\mathbf{s}} \in \{0, 1, 2\}^d$. $\mathbf{B}$ is the public parameter of the commitment scheme Π_{Com} (see Appendix A.4) and $\mathbf{b}_i^\top$ denotes its row vector. In case abort occurs, we send $\perp$ as the fifth message

the hardness of the primal attack and assess the difficulty of the (ternary secret) MLWE and DSMR problem required by the extractable LinHC protocol. We observe that we would require a root Hermite factor of at least 1.0045 for our parameters, which is believed to provide 128-bits of post-quantum security, e.g., [BLS19, ESLL19, ESS+19].

C Recap on Unruh's Transform

In this section, we provide a minimal recap on the Unruh transform [Unr15]. Moreover, we provide intuition and a concrete example on why applying the transform on lattice-based Σ-/public-coin HVZK interactive protocols with a large challenge set may incur a large overhead compared to simply using the Fiat-Shamir transform.

C.1 High Level Idea of Unruh's Transform

One of the difficulties of proving security of the Fiat-Shamir transform in the QROM is largely because the reduction algorithm cannot observe what the quantum adversary is querying to the QRO.[27] The main idea

[27] Note that in recent years, techniques that allow observing the adversary's queries as in the classical ROM (in some situation) have emerged so we now know how to prove the Fiat-Shamir transform in the QROM under appropriate conditions, e.g., [DFMS19, LZ19].

$\mathsf{NIZK.Prove}^{\mathsf{H}}(\mathsf{crs} = \mathbf{B}, \mathsf{X} = (\mathbf{A}, \mathbf{u}), \mathsf{W} = s)$

1: **for** $i \in [t]$ **do**
2: $\quad y_i \leftarrow R_q$
3: $\quad \mathbf{w}_i \leftarrow \mathbf{A}\hat{\mathbf{y}}_i \in \mathbb{Z}_q^m$
4: $\quad \mathbf{e}_i = (e_{1,i}, e_{2,i}, e_{3,i}, e_{4,i}, e_{5,i}, e_{6,i})^\top \leftarrow S^6_{B_\mathbf{e}}$
5: $\quad t_{1,i} \leftarrow \mathbf{b}_1^\top \mathbf{e}_i$
6: $\quad t_{2,i} \leftarrow \mathbf{b}_2^\top \mathbf{e}_i + y_i$
7: $\quad t_{3,i} \leftarrow \mathbf{b}_3^\top \mathbf{e}_i + s$
8: $\quad t_{4,i} \leftarrow \mathbf{b}_4^\top \mathbf{e}_i + y_i(2s-3)$
9: $\quad t_{5,i} \leftarrow \mathbf{b}_4^\top \mathbf{e}_i + y_i^2(s-3)$
10: $\quad \mathbf{t}_i \leftarrow (t_{1,i}, t_{2,i}, t_{3_i}, t_{4,i}, t_{5,i})^\top$
11: $(c_i)_{i\in[t]} \leftarrow \mathsf{H}\big(0 \| (\mathbf{t}_i, \mathbf{w}_i)_{i\in[t]}\big)$
12: **for** $i \in [t]$ **do**
13: $\quad z_{0,i} \leftarrow c_i \cdot s + y_i$
14: $\quad$ **for** $j \in \{(i-1)t'+1, \cdots, (i-1)t'+t'\}$ **do**
15: $\quad\quad \mathbf{r}_j \leftarrow D^6_{\phi\cdot\mathbf{B}_\mathbf{r}}$
16: $\quad\quad x_{0,j} \leftarrow \mathbf{b}_1^\top \mathbf{r}_j$
17: $\quad\quad x_{1,j} \leftarrow (\mathbf{b}_2^\top + c_i \mathbf{b}_3^\top)\mathbf{r}_j$
18: $\quad\quad x_{2,j} \leftarrow \big((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot \mathbf{b}_3^\top - z_{0,j} \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top\big)\mathbf{r}_j$
19: $(\beta_j)_{j\in[tt']} \leftarrow \mathsf{H}\big(1 \| ((\mathbf{t}_i, \mathbf{w}_i)_{i\in[t]}), (c_i)_{i\in[t]}, \big((z_{0,i})_{i\in[t]}, (x_{0,j}, x_{1,j}, x_{2,j})_{j\in[tt']}\big)\big)$
20: **for** $j \in [tt']$ **do**
21: $\quad \mathbf{z}_j \leftarrow \beta_j \cdot \mathbf{e}_{\lceil j/t' \rceil} + \mathbf{r}_j$
22: **for** $k \in [t']$ **do**
23: $\quad \bar{\mathbf{z}}_k \leftarrow [\mathbf{z}_k \| \mathbf{z}_{t'+k} \| \cdots \| \mathbf{z}_{(t-1)t'+k}]$
24: $\quad \bar{\mathbf{e}}_k \leftarrow [\beta_k \mathbf{e}_1 \| \beta_{t'+k} \mathbf{e}_2 \| \cdots \| \beta_{(t-1)t'+k} \mathbf{e}_t]$
25: $\quad b \leftarrow \mathsf{Rej}(\bar{\mathbf{z}}_k, \bar{\mathbf{e}}_k, \phi, B_\mathbf{r}, \mathsf{err})$
26: $\quad$ **if** $b = \perp$ **then**
27: $\quad\quad$ **goto** Line 1
28: **return** $\pi := \big((\mathbf{t}_i, c_i, z_{0,i})_{i\in[t]}, (\beta_j, \mathbf{z}_j)_{j\in[tt']}\big)$

Figure 17: Prover algorithm of Bootle et al.'s [BLS19] exact sound NIZK for the relation $\mathcal{R}_{\mathsf{ES}}$in the CRS/ROM model. The statement $\mathsf{X} = (\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{m\times d} \times \mathbb{Z}_q^m$ and witness $s \in R_q$ satisfies $\mathbf{A}\hat{\mathbf{s}} = \mathbf{u}$ and $\hat{\mathbf{s}} \in \{0,1,2\}^d$. $\mathbf{B} \in R_q^{5\times 6}$ is the public parameter of the (implicit) commitment scheme Π_{Com} (see Appendix A.4), and $\mathbf{b}_i^\top$ and $b_{i,j}$ denotes its i-th row vector and (i,j)-th entry, respectively.

behind Unruh's transform is to bypass this issue by forcing the prover to compute all the responses for each challenge and then committing them with an extractable commitment scheme (which exists unconditionally in the QROM).

Let us explain the construction in more detail. Assume the Σ-protocol has three moves: the prover first sends the commitment α, the verifier outputs a random challenge $\beta \in \mathsf{ChSet}$, and the prover sends the response γ. To turn this into an NIZK, the prover first generates α and then creates the response γ for all $\beta \in \mathsf{ChSet}$. That is, it generates $\{(i, \gamma_i)\}_{i\in\mathsf{ChSet}}$. The prover then commits to each (i, γ_i) as com_i using the extractable commitment scheme and creates a new challenge $\mathsf{ch} = \mathsf{H}(\alpha, \{\mathsf{com}_i\}_{i\in\mathsf{ChSet}})$. For this challenge ch, the prover provides the commitment randomness $\mathsf{rand}_{\mathsf{ch}}$ to the commitment $\mathsf{com}_{\mathsf{ch}}$. Finally, the prover outputs the proof $\pi = (\alpha, \{\mathsf{com}_i\}_{i\in\mathsf{ChSet}}, (\mathsf{rand}_{\mathsf{ch}}, \gamma_{\mathsf{ch}}))$. On input the proof $\pi = (\alpha, \{\mathsf{com}_i\}_{i\in\mathsf{ChSet}}, (\mathsf{rand}, \gamma))$, the verifier computes $\mathsf{ch}' = \mathsf{H}(\alpha, \{\mathsf{com}_i\}_{i\in\mathsf{ChSet}})$ and checks if $\mathsf{com}_{\mathsf{ch}'} = \mathsf{Com}(\gamma; \mathsf{rand})$ and whether $(\alpha, \mathsf{ch}', \gamma)$ is a valid transcript for the underlying Σ-protocol.

So why should this be secure in the QROM? With a simple cut-and-chose argument, we can run the above protocol several times to be sure that the adversary committed to at least two valid responses in one of the runs. That is, there exists two com_i and com_j for $i \neq j \in \mathsf{ChSet}$ such that they are both commitments to responses γ_i and γ_j where (α, i, γ_i) and (α, j, γ_j) are valid transcripts. Therefore, the reduction algorithm

$\mathsf{NIZK.Verify}^{\mathsf{H}}(\mathsf{crs}, \mathsf{X}, \pi)$

$\left((\mathbf{t}_i, c_i, z_{0,i})_{i\in[t]}, (\beta_j, \mathbf{z}_j)_{j\in[tt']}\right) \leftarrow \pi$
for $i \in [t]$ **do**
  $\mathbf{w}_i \leftarrow \mathbf{A}\hat{\mathbf{z}}_{0,i} - c_i \cdot \mathbf{u}$
  for $j \in \{(i-1)t'+1, \cdots, (i-1)t'+t'\}$ **do**
    $x_{0,j} \leftarrow \mathbf{b}_1^\top \mathbf{z}_i - \beta_j \cdot t_{1,i}$
    $x_{1,j} \leftarrow (\mathbf{b}_2^\top + c_i \mathbf{b}_3^\top)\mathbf{z}_j - \beta_j \cdot (c_i \cdot t_{3,i} + t_{2,i})$
    $x_{2,j} \leftarrow \left((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot \mathbf{b}_3^\top - z \cdot \mathbf{b}_4^\top + \mathbf{b}_5^\top\right)\mathbf{z}_j - \beta_j \cdot ((z_{0,i} - c_i)(z_{0,i} - 2c_i) \cdot t_{3,i} - z_{0,i} \cdot t_{4,i} + t_{5,i})$
$(c'_i)_{i\in[t]} \leftarrow \mathsf{H}\big(0 \| (\mathbf{t}_i)_{i\in[t]}, (\mathbf{w}_i)_{i\in[t]}\big)$
$(\beta_j)_{j\in[tt']} \leftarrow \mathsf{H}\big(1 \| ((\mathbf{t}_i, \mathbf{w}_i)_{i\in[t]}), (c_i)_{i\in[t]}, \big((z_{0,i})_{i\in[t]}, (x_{0,j}, x_{1,j}, x_{2,j})_{j\in[tt']}\big)\big)$
for $j \in [tt']$ **do**
  if $\|\mathbf{z}_j\|_2 > B_{\mathbf{z}}$ **then**
    return $\perp$
if $(c_i)_{i\in[t]} \neq (c'_i)_{i\in[t]} \ \vee \ (\beta_j)_{j\in[tt']} \neq (\beta'_j)_{j\in[tt']}$ **then**
  return $\perp$
else
  return $\top$

Figure 18: Verifier algorithm of Bootle et al.'s [BLS19] exact sound NIZK for the relation $\mathcal{R}_{\mathsf{ES}}$in the CRS/ROM model.

only needs to run the adversary once and can directly extract two valid transcripts from the proof π (using the extractability of the commitment scheme), thus bypassing the difficulty of handling QROs. Finally, once two valid transcripts are obtained, the reduction algorithm runs the special soundness extractor of the underlying Σ-protocol to extract the witness.

Although the concrete analysis requires more care, Unruh showed that the above blueprint results in a QROM secure NIZK. The upside of this transform is that it works for any Σ-protocol, it is straight-line extractable, and the proof is tight. We next explain some of the downside.

C.2 Two Reasons for Inefficiency

There is two source of possibly inefficiency of the Unruh transform. First, notice that the above idea crucially relies on the fact the prover can run over all the challenge set ChSet. Therefore, the Unruh transform intrinsically requires ChSet of the underlying Σ-protocol to be polynomially small. In case the Σ-protocol already relies on a small challenge space, e.g., [Ste94, KTX08, CDG+17], this is not so much of an issue. However, this is clearly an issue when the Σ-protocol has a large challenge space. In such a case, we would have to downgrade the Σ-protocol to only use a small subset of the challenge space and then perform parallel repetition to amplify the soundness error to be negligible. For instance, consider a Σ-protocol with challenge set size 2^{256} (for 128-bits of quantum security) with soundness error 2^{-128}. Then, to apply the Unruh transform, we first restrict the challenge set size to be small, say 2^8. This would increase the soundness error to 2^{-4} so we need to run the underlying Σ-protocol at least 32-times to boost its soundness error to 2^{-128}. This leads to at least a factor of 32 blowup. As a rule of thumb, if the number of parallel repetition is T, then the number of challenge set can be set as $2^{256/T}$. (See [Unr15, Def. 13 and Thm. 18] for more details.)

The second source of possible ineffeciency is the additional commitments $\{\mathsf{com}_i\}_{i\in\mathsf{ChSet}}$ that must be sent in the proof π. Since we need the commitments to be extractable, these are at least as large as the response γ. Combined with the above, a naive calculation shows that the resulting NIZK has proof size around $T \cdot (|\alpha| + 2^{256/T} \cdot |\gamma|)$. Rewriting this by using the challenge set size $M = 2^{256/T}$ instead, we get $\frac{256}{\log M} \cdot (|\alpha| + (M+1) \cdot |\gamma|)$. In contrast, if the underlying Σ-protocol had challenge set size N, then the transcript size is roughly $\frac{256}{\log N} \cdot (|\alpha| + |\gamma|)$. It is clear that when N is large and M is small, the overhead is large. For example, in case $|\alpha| \approx |\gamma|$, $N = 2^{256}$, and $M = 2^4$, the proof overhead is roughly 1000. In contrast,

when N is already small, say $N = 2$ or 3, then the overhead is reasonable. For instance, starting with a proper Σ-protocol and through optimization, the Unruh transform can lead to truly practical scheme as demonstrated in [CDG+17].

C.3 Applying the Unruh Transform to Bootle et al's Protocol

We provide the concrete details on the Unruh transform applied to Bootle et al's 5-round public-coin HVZK interactive exact sound proof [BLS19] in Section 5. We use the same concrete application of proving knowledge of the ternary secret in LWE samples over $\mathbb{Z}_q$ considered in Appendix B.2 to compare our QROM secure NIZK and Bootle et al's classical ROM secure NIZK. Here, since the Unruh transform only applies to Σ-protocols, we rely on the extended Unruh transform proposed by Chen et al. [CHR+18] that works for 5-round public-coin HVZK interactive protocol with the second challenge set restricted to be *binary*. That is, the verifier outputs a random challenge in $\{0, 1\}$ in the fourth round.

In Bootle et al's and our protocols, the first and second challenge set sizes are $q = 2^{32}$ (modulus size) and $d = 2^{11}$ (LWE parameter), respectively. To apply the extended Unruh transform, the second challenge set size needs to be binary. In addition, the first challenge set size needs to be much smaller than 2^{32} since creating 2^{32}-commitments is impractical. To provide an estimate, let us set the first challenge set size to be M. Then, similarly to the discussion in Section 5, we perform $t' = \lceil \log M \rceil$ lower repetition so that $M = |\{0, 1\}|^{t'}$ and perform $t = \lceil \frac{256}{\log M} \rceil$ upper repetitions to reach the 128-bit of post-quantum security level. Then, the Unruh transform requires the prover to run through the first M-challenges and generate and commit to all the M-responses each having size $|z_{0,i}| = d\lceil \log q \rceil / 8$ bytes (see Figure 17). It further requires the prover to run through the first M and second binary challenges and generate and commit to all the $2M$-responses each having size $|\mathbf{z}_j| = 6d\lceil \log 12\phi B_{\mathbf{r}} \rceil$ bytes (see Figure 17). Therefore, asymptotically, we require a total of $(6td\lceil \log q \rceil + 6tt'd\lceil \log 12\phi B_{\mathbf{r}} \rceil + t\lceil \log q \rceil + tt'\lceil \log 2d \rceil)/8 + X$ bytes, where $X = tMd(\lceil \log q \rceil + 12t'\lceil \log 12\phi B_{\mathbf{r}} \rceil)/8$ bytes is the additional commitments required by the Unruh transform. Here, recall M is some polynomial larger than 3 (for the security proof), and $\lceil \frac{256}{\log M} \rceil$ and $t' = \lceil \log M \rceil$. It turns out that the total proof size is minimized when setting $M = 4$ and becomes 44.9 MB, which is roughly a factor 51.8 larger than Bootle et al's protocol that has proof size 812KB. Here, it may seem the overhead incurred by the Unruh transform is smaller than expected. The main reason for this is that Bootle et al's protocol already relied on parallel repetition to obtain negligible soundness error since the first and second challenge set sizes are only $q = 2^{32}$ and $d = 2^{11}$, respectively.