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

Shuichi Katsumata

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.

1 Introduction

The Fiat-Shamir transform [24] 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 [8] 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

[1] We may simply refer to NIZK proofs or NIZK proofs of knowledge as NIZKs when the distinction is not relevant.

but not limited to [8,50,45,9,46,47,31,51,18,35,17], 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 [3,2,10,48,21,1], based on the techniques developed by Lyubashevsky [37,38] 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 [46] but incurs a large overhead, say at least $\times 50$, 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 [24] and the other is the Unruh transform [46].

Recently, Don et al. [18] and Liu and Zhandry [35] 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 [35] (and partially in [18]) 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 [32,30], having a tighter reduction is desirable.

On the other hand, Unruh [46] 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 [5] and anonymous attestations [11]. On the other

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

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., [13], it becomes prohibitively large as the challenge set grows. Recently, Chen et al. [14] 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 [37,38] and the other builds on Stern's protocol [44,29]. While the QROM security of the latter approach is well understood since it has a simple combinatorial "commit-and-open" structure [18,17], the QROM security of the former approach remains elusive. Notably, for the recent lattice-based protocols such as [3,2,10,48,21,1], 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 [24], while also providing a tight reduction along with a straight-line extractor like the Unruh transform [46].

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 [3,10,48,21,1] into a QROM secure NIZK that is also straight-line (simulation) extractable [23]. 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 [1] 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 [10] into a QROM secure NIZK by only increasing the proof size by a factor of 2.6.[3] This is in contrast to using

[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 [31].

the recent extended Unruh transform [14][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 Sec. 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 Sec. 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 Sec. 4.1)
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 Sec. 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 [36,43] defined over a module NTRU lattice [15]. (See Sec. 3.4)
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 [10] 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 [3,48,21,1]. (See Sec. 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 [25] and Unruh [46]) 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

[4] Since Bootle et al's protocol requires slightly more transcripts for special soundness compared to those considered in [14], the security proof of [14] 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.

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 [25] 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. 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.

Related works on Σ-protocols, NIZKs, and lattice-based ZK proofs and QROM secure signatures are provided in the full version.

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 [37,38], 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 $\mathbf{Ae} = \mathbf{u}$.[6] 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 construct an extractor SL-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 $\delta_{\mathbf{e}}$

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

and $\delta_{\mathbf{r}}$ are commitment randomness.[7] 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; $\mathbf{A}\mathbf{z} = \beta \cdot \mathbf{u} + \mathbf{w}$ 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

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

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 [25,46].[8]

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.

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 Sec. 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 [18,35] 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 $\mathsf{SL\text{-}Extract}$ 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

[8] To be precise, [25] 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.

proof for the deterministic signature scheme based on the Fiat-Shamir transform (which captures the essence of a simulation sound/extractable NIZK).[9]

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. [31] (which themselves follow [46,47]) 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* [8]. 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. [31] 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 Sec. 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, the extractor probes

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

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 [49,28,31].

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 [41,26] 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 [37,38] 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.[10] 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. [10] 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

[10] 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.

"collapsingness" [35,18]. We can instead try applying the recent Unruh transform extended to 5-round protocols by Chen et al. [14] by limiting the second challenge used by the verifier to be binary. For completeness, we show in the full version 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 Sec. 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 [3,48,21,1]. 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 Dilithium proposed in [31]. Roughly, this is because when viewed as an NIZK, ours achieve a stronger property: while [31] only achieves soundness, we also achieve (straight-line) proof of knowledge.

2 Preliminary

The notations we use in this paper and a minimal set of tools on quantum computation in provided in the full version.

2.1 Σ-Protocol

We use the standard notion of Σ-protocol in the *common reference string* model.[11] We note that it is standard in lattice-based protocols to consider *non-abort* honest-verifier zero-knowledge (naHVZK), where the ZK simulator is only required to simulate non-aborting transcripts. Due to page limitation, we refer the basic definitions to the full version and only provide the definition of *straight-line proof of knowledge* below.

Definition 2.1 (Straight-line proof of knowledge). *A Σ-protocol has a (quantum) ϵ_{IndO}-*straight-line proof of knowledge (SL-PoK) *if there exists a PPT simulator* SimSetup *and a PPT straight-line extractor* SL-Extract *with the following properties:*

- *For any QPT $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{IndCRS}}(\mathcal{A})$ defined below is less than $\epsilon_{\mathsf{IndCRS}}$:* $\mathsf{Adv}^{\mathsf{IndCRS}}(\mathcal{A}) := |\Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) : \mathcal{A}(1^\kappa, \mathsf{crs}) \to 1] - \Pr[(\widetilde{\mathsf{crs}}, \tau) \leftarrow \mathsf{SimSetup}(1^\kappa) : \mathcal{A}(1^\kappa, \widetilde{\mathsf{crs}}) \to 1]|$.
- *For any QPT $\mathcal{A}$ and any $\mathsf{X} \in \mathcal{L}$ satisfying*

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

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

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.*[12] *Concretely, if ϵ is non-negligible and $|\mathsf{ChSet}|$ is super-polynomially large, then* SL-Extract *runs in polynomial time.*

2.2 Lattices

Basic notations and well known tools for lattices are provided in the full version. We let S_η denote the set of all elements in $a\in R_q$ such that $\|w\|_\infty\leq\eta$. As with all Σ-protocols that rely on the Fiat-Shamir with abort technique, we use the rejection sampling technique [37,38]. We denote the rejection sampling algorithm as Rej. To construct extractable LinHC protocols, we rely on a variant of the standard module learning with errors MLWE assumption, where the adversary is allowed to obtain a superposition of *independent* MLWE samples (which remains as hard as the standard MLWE assumption). We also consider the quantum accessible decisional small *matrix* ratio (DSMR) assumption, which is essentially the underlying hardness assumption of (module) NTRU.

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[13] in the QROM from a possibly non-quantum secure Σ-protocol (see Sec. 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 Sec. 4.1).

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

[13] 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 [23].

3.1 Definition

An illustration of the extractable LinHC protocol is provided in Figure 1. 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: $\left(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r}\right)$		Verifier: $\mathsf{K}_{\mathsf{com}}$
$(\mathsf{com}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{K}_{\mathsf{com}}, (\mathbf{e}_i)_{i\in[N]}, \mathbf{r})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\boldsymbol{\beta} = (\beta_i)_{i\in[N]}}$	$\boldsymbol{\beta} \leftarrow \mathsf{ChSet}$
$\mathbf{z} = \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}$ $\mathsf{op} \leftarrow \mathsf{Open}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, \mathbf{z}), \mathsf{st}))$	$\xrightarrow{(\mathbf{z}, \mathsf{op})}$	
		$\mathsf{Verify}(\mathsf{K}_{\mathsf{com}}, (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))) \stackrel{?}{=} \top$

Fig. 1. 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 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:*

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* $\mathbf{z} \leftarrow \sum_{i=1}^{N} \beta_i \cdot \mathbf{e}_i + \mathbf{r}$[14]*, 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.

[14] 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., [21].

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* $\mathsf{op} \neq \bot$.
- *The probability that an honestly generated transcript* $(\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op}))$ *contains* $\mathsf{op} = \bot$ *is bounded by* δ_1. *In particular,* $\Pr[\mathsf{op} = \bot] \leq \delta_1$ *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:*

- $D^{\mathcal{L}}_{\mathsf{trans}}(\rho, (\mathbf{e}_i)_{i\in[N]})$: *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* $\mathsf{op} \neq \bot$ *(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* $(\mathbf{r}, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))$.

In above, we assume $D^{\mathcal{L}}_{\mathsf{trans}}$ *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, *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}:

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

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 4 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}:*

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

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 $S_f(\rho, \mathsf{com})$ as*

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

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 that help simplify the proofs in some cases. The first property is useful when the underlying public-coin HVZK interactive protocol already uses a small (i.e., poly-large) challenge set. These shows up in multi-round protocols where the verifier may sample randomness from different challenge sets in each round. (See Sec. 5 for an example.) The second property allows to argue that for each challenge $\boldsymbol{\beta} \in \mathsf{ChSet}$, there exist at most one response $\mathbf{z}$ that passes the verification. Due to page limitation, we omit the details to the full version.

3.2 Simplified Definition of Extractable LinHC

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. We omit the details to the full version.

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 Sec. 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 $\|\mathbf{z}\|_2 \leq B$.[15] 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 $(\mathbf{e}_i)_{i\in[N]}$.

3.4 Construction of Extractable LinHC

We propose two constructions of extractable LinHC protocols: one based only on MLWE and the other based on MLWE *and* DSMR. Since the two constructions are almost identical, we explain the former and refer the details on the latter to the full version. The latter has proof size half of the former while relying on the extra DSMR assumption. The construction of our first extractable LinHC protocol based on MLWE is provided in Figure 2.

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

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

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

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

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

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

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

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

Fig. 2. An extractable LinHC protocol based on MLWE.

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

Parameters and asymptotic size. Let the dimension d of the ring R_q be larger than 256 and n, m be positive integers such that $n \leq m$,[16] $p < q$ 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, 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 $\boldsymbol{\beta}$. Then, the total size becomes $2mdN\log q + (n+2m)d \cdot \log(10\phi T) + |\mathsf{ChSet}|$.

Properties. Due to page limitation, we omit the details of the proof of correctness and the quantum accessible non-abort HVZK ($\mathsf{QAnaHVZK}$) to the full version. We note that for $\mathsf{QAnaHVZK}$, we rely on the *quantum accessible* MLWE assumption.

$\underline{\widetilde{\mathsf{H}}(\rho)}$
1: $(\rho_1, \rho_2, \rho_3) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
2: $\mathbf{A} \leftarrow R_q^{m\times n}[\rho_1]$
3: $(\mathbf{D}_1, \mathbf{D}_2) \leftarrow S_\eta^{m\times m}[\rho_2] \times S_\eta^{m\times n}[\rho_3]$
4: $\mathbf{B} \leftarrow \mathbf{D}_1\mathbf{A} + \mathbf{D}_2$
5: **return** $(\mathbf{A}, \mathbf{B})$

$\underline{\mathsf{SimOracle}(1^\kappa)}$
1: $\mathsf{K} \leftarrow \mathcal{K}$ ▷ Sample PRF key
2: **return** $(\widetilde{\mathsf{H}}, \tau := \mathsf{K})$

$\underline{\mathsf{LinCExtract}(\tau = \mathsf{K}, \rho, \mathsf{trans} = (\mathsf{com}, \boldsymbol{\beta}, (\mathbf{z}, \mathsf{op})))}$
1: $(\rho_1, \rho_2, \rho_3) \leftarrow \mathsf{PRF}(\mathsf{K}, \rho)$
2: $\mathbf{D}_1 \leftarrow S_\eta^{m\times m}[\rho_2]$
3: $\left((\mathbf{t}_{i,1}.\mathbf{t}_{i,2})_{i\in[N]}, \mathbf{w}_1, \mathbf{w}_2\right) \leftarrow \mathsf{com}$
4: $(\beta_1, \cdots, \beta_N) \leftarrow \boldsymbol{\beta}$
5: $(c, L) \leftarrow (0, (\boldsymbol{\beta}, \mathbf{z}))$
6: **while** $|L| \leq k \ \vee \ c \leq T^*$ **do**
7: $\quad \widetilde{\boldsymbol{\beta}} = (\widetilde{\beta}_1, \cdots, \widetilde{\beta}_N) \leftarrow \mathsf{ChSet} \backslash L_{\boldsymbol{\beta}}$
8: $\quad \widetilde{\mathbf{z}} \leftarrow \left(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2\right)$
9: $\qquad -\mathbf{D}_1\left(\sum_{i=1}^{N} \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1\right) \mod p$
10: $\quad$ **if** $f(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}}) = \top$ **then** $L \leftarrow L \cup \{(\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{z}})\}$
11: $\quad c \leftarrow c + 1$
12: **if** $|L| < k$ **then return** $\perp$
13: **else return** L

Fig. 3. Description of $\mathsf{SimOracle}$, $\widetilde{\mathsf{H}}$, and $\mathsf{LinCExtract}$ for the extractable LinHC protocol in Figure 2. 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 3.1 ($\mathcal{F}_B$-Almost straight-line extractable). *Assume $B \geq \sqrt{2nd} \cdot \phi \cdot T$, $2\sqrt{2}p(nd\eta + \sqrt{nmd}\eta + \sqrt{nd})\phi T + 2B < q/2$, and $B \leq (p-1)/4$. Define the oracle simulator $\mathsf{SimOracle}$ and linear commitment extractor $\mathsf{LinCExtract}$ as in Figure 3, where T^* in Line 6 of algorithm $\mathsf{LinCExtract}$ is $T^* = \frac{k\cdot\delta\cdot|\mathsf{ChSet}|}{|S_f(\rho,\mathsf{com})|-k}$. Then,*

[16] 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.

the extractable LinHC *protocol in Figure 2 is* $\mathcal{F}_B$*-almost straight-line extractable. 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*

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

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

Proof. We only prove Item 2 below and refer the others to the full version.

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 3. Therefore, since $\mathbf{v} := (\sum_{i=1}^N \widetilde{\beta}_i \cdot \mathbf{t}_{i,2} + \mathbf{w}_2) - \mathbf{D}_1(\sum_{i=1}^N \widetilde{\beta}_i \cdot \mathbf{t}_{i,1} + \mathbf{w}_1) = p \cdot (\mathbf{D}_2\widetilde{\mathbf{z}}_1 - \mathbf{D}_1\widetilde{\mathbf{z}}_2 + \widetilde{\mathbf{z}}_3) + \widetilde{\mathbf{z}}$, we have

$$\begin{aligned}\|\mathbf{v}\|_\infty \leq& p \cdot \left(\sqrt{nd}\|\mathbf{D}_2\|_\infty \cdot \|\widetilde{\mathbf{z}}_1\|_2 + \sqrt{md}\|\mathbf{D}_1\|_\infty \cdot \|\widetilde{\mathbf{z}}_2\|_2 + \|\widetilde{\mathbf{z}}_3\|_\infty\right) + \|\widetilde{\mathbf{z}}\|_\infty \\ \leq& \sqrt{2}p(nd\eta + \sqrt{nm}d\eta + \sqrt{nd})\phi T + 2B < q/2,\end{aligned}$$

where we have $\|\widetilde{\mathbf{z}}\|_2 \leq B$ by definition of $\mathcal{F}_B$ (see Sec. 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 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 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. □

We note that we can get an asymptotically more efficient extractor by allowing algorithm $\mathsf{LinCExtract}$ to be QPT and perform Grover's search. Finally, we also discuss how to "downgrade" the above extractable LinHC protocol to only satisfy the properties of a simplified/classical extractable LinHC protocol. The benefit of doing this is that it provides tighter reductions since we no longer need to work with QROs. The details are provide in the full version.

4 How to Use Extractable LinHC

In this section, we provide a basic example of bootstrapping the ROM secure Lyubashevsky's Σ-protocol [37,38] 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 Sec. 5, we see how these ideas can be used to prove QROM security of more complex protocols.

As explained in the beginning of Sec. 3, we can either construct a (1) quantum straight-line extractable Σ-protocol using the simplified extractable LinHC protocol (see Sec. 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 ⇒ 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. 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. 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 $B'_{\mathbf{e}} > B_{\mathbf{e}}$. It is known that the Σ_{Lyu}-protocol is naHVZK and satisfies relaxed 2-special soundness.

Quantum secure Σ-protocol. The construction is depicted in Figure 4. Setup of the Σ protocol runs KeyGen of the extractable LinHC protocol. Below, we show correctness, naHVZK, and SL-PoK of our Σ-protocol in Figure 4. Since the first two properties follows almost immediately from the underlying Σ_{Lyu}-protocol and the simplified extractable LinHC protocol, we omit them to the full version.

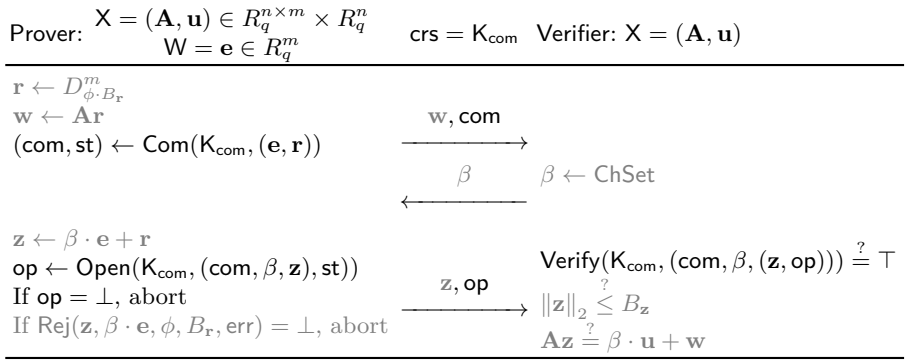

Fig. 4. Quantum secure Σ-protocol in the CRS model for the lattice relation $\mathbf{A}\mathbf{e} = \mathbf{u}$, where crs is $\mathsf{K}_{\mathsf{com}} \leftarrow \mathsf{KeyGen}(1^\kappa)$. The witness $\mathbf{e}$ satisfies $\|\mathbf{e}\|_2 \leq B_{\mathbf{e}}$. The gray indicates the components that are used in the Σ_{Lyu}-protocol.

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

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 it 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** $\mathsf{W} := \mathbf{z}^*$

Fig. 5. Description of SimSetup and SL-Extract for the Σ-protocol in Figure 4.

Lemma 4.1 (SL-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 SimKeyGen and linear commitment extractor 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 2.1, 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 4 is a straight-line PoK *with simulator* SimSetup *and straight-line extractor* SL-Extract *described in Figure 5.*

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 2.1. Then, we have

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

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 better readability, we can rewrite the l.h.s of Equation (1) as

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

Here, $f \in \mathcal{F}_{B_{\mathbf{z}}}$ is the function that on input $(\beta, \mathbf{z})$, outputs $\top$ if and only if $\|\mathbf{z}\|_2 \leq B_{\mathbf{z}}$ and $\mathbf{A}\mathbf{z} = \beta \cdot \mathbf{u} + \mathbf{w}$, where $\mathbf{w}$ is the vector included in α output by $\mathcal{A}$. 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 $\Pr[\mathsf{Verify}(\widetilde{\mathsf{crs}}, \mathsf{X}, (\alpha, \beta, \gamma)) = \top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com})| < \Gamma] < \frac{\Gamma}{|\mathsf{ChSet}|} = \frac{\epsilon - \epsilon_{\mathsf{IndCom}}}{2}$. Combining this with Equations (1) and (2), we have $\Pr[\mathsf{Verify}(\widetilde{\mathsf{crs}}, \mathsf{X}, (\alpha, \beta, \gamma)) = \top \ \wedge \ |S_f(\widetilde{\mathsf{K}}_{\mathsf{com}}, \mathsf{com})| \geq \Gamma] \geq \frac{\epsilon - \epsilon_{\mathsf{IndCom}}}{2}$. 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, 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 5, with probability at least $\frac{\epsilon - \epsilon_{\mathsf{IndCom}}}{2} \cdot (1 - 2^{-\kappa})$, SL-Extract moves on to Line 3. By 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. This completes the proof. □

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 eu-cma secure signature scheme using the Fiat-Shamir transform The main technicality of this section is to show that even if an adversary gets to observe polynomially many simulated proofs (i.e., signatures), we are still able to extract a witness from a valid proof (i.e., extract the secret key from a signature forgery) output by the adversary *without* rewinding.

QROM secure signature scheme. The construction of our (deterministic) signature scheme in the QROM is provided in Figure 6.[17] 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 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 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.

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

1: $(\mathbf{A}, \mathbf{e}) \leftarrow R_q^{n\times m} \times S^m_{B_\mathbf{e}}$
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: $\quad \mathbf{r} \leftarrow D^m_{\phi \cdot B_\mathbf{r}}[\rho_\mathbf{r}]$
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** $\sigma := (\beta, \mathbf{z}, \mathsf{com}, \mathsf{op})$

Fig. 6. QROM secure signature scheme by applying the Fiat-Shamir transform to our Σ-protocol in Figure 4. Oracles $\mathsf{H}_{\mathtt{LHC}}$ and $\mathsf{H}_{\mathtt{FS}}$ are implemented using H.

Properties. Due to page limitation, we provide the proof of $\mathsf{eu\text{-}cma}$ security in the full version. For an overview of the proof, we refer the readers to the technical overview in Sec. 1.2. The main technicality of the proof is showing that with high probability, there must have been another challenge the adversary was able to forge on even though it had some control over which challenge it used through quantumly accessing the random oracle $\mathsf{H}_{\mathtt{FS}}$.

[17] 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 [31] for example), we ignore this unrelated subtlety for better readability.

5 Application: Quantum Secure 5-Round Public-Coin Exact Sound Proof and 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 [10]. The main motivation for choosing [10] as the case study is because the ideas presented in this section can be directly applied to other recent works [3,21,48,1]. We can convert the protocol of [10] into either (1) a quantum secure straight-line extractable *interactive* proof using the simplified extractable LinHC protocol (as in Sec. 4.1) or (2) into a quantum secure simulation straight-line extractable NIZK (or a signature scheme) using the extractable LinHC protocol (as in Sec. 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 [10] 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). 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 $\mathcal{R}_{\mathsf{ES}} = \{s \in R_q \mid \mathbf{A}\hat{\mathbf{s}} = \mathbf{u} \ \wedge \ \hat{\mathbf{s}} \in \{0,1,2\}^d\}$. 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. 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.

Quantum secure exact sound protocol. The protocol is depicted in Figure 7. 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 4). Correctness and naHVZK are straightforward to prove and we omit them to the full version.

The high level idea of the proof for straight-line proof of knowledge is similar to those provided by Bootle et al. [10, 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.

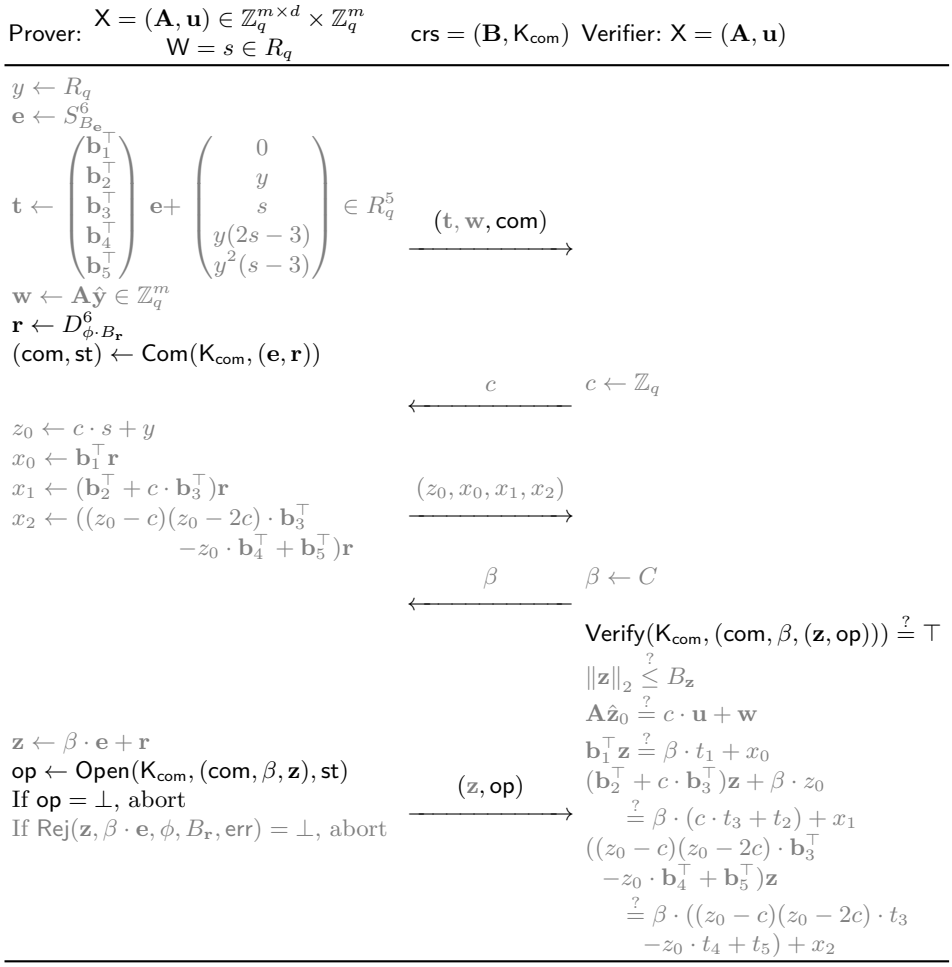

Fig. 7. Quantum secure exact sound public-coin interactive protocol in the CRS model for the relation $\mathcal{R}_{\mathsf{ES}}$. $\mathbf{B} \in R_q^{5\times 6}$ is the public parameter of the (implicit) commitment scheme Π_{Com} Π_{Com}, and $\mathbf{b}_i^\top$ denotes its i-th row vector. The gray indicates the components that are used in the protocol of Bootle et al. [10].

Lemma 5.1 (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* $\|\mathbf{z}\|_2 \leq B_{\mathbf{z}}$.

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* $\epsilon > 3/q + 2/d$[18] *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* $(\epsilon - \nu)/3$ *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 $\mathsf{trans}^* = \big((\mathbf{t}, \mathbf{w}, \mathsf{com}), c^{(1)}, (z_0^{(1)}, x_0^{(1)}, x_1^{(1)}, x_2^{(1)}), \beta^{(1,1)}, (\mathbf{z}^{(1,1)}, \mathsf{op}^{(1,1)})\big)$. Firstly, since $\mathcal{A}$ has advantage greater than $3/q + 2/d$, using the same statistical argument made in the proof of Lemma 4.1, with probability at least $1/3$, the transcript trans^* output by $\mathcal{A}$ satisfies 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 $(k, j) \in [3] \times [2]$. Below, we first show how SL-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.

We define 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 $(\epsilon - \epsilon_{\mathsf{IndCom}})/3$ in outputting a valid transcript trans^* with the above property run on this modified crs. Next, SL-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 $|C| = d$[19], 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 $\|\mathbf{z}'\|_2 \leq B_{\mathbf{z}}$. That is, all the extracted β satisfies $\beta \in S_f(\mathsf{K}_{\mathsf{com}}, \mathsf{com})$. Moreover, 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, w, x_1, x_2)).[20] 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,

[18] Bootle et al. [10, Theorem 3.1] only requires $\epsilon > 2/q + 2/d$. However, this slight modification makes our proof slightly easier to state and has minimal impact on the concrete efficiency of the scheme.

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

[20] This argument relies on a natural yet extra property of the LinHC. The detail is provided in the full version.

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 $\mathbf{t}$. 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 with $\|\overline{\mathbf{z}}_t\|_2 \leq 2B_{\mathbf{z}}$. Hence, unless $\mathcal{A}$ breaks the binding property of the commitment, 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 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

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

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 and making 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

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

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 $(c^{(k)}, (z_0^{(k)}, x_0^{(k)}, x_1^{(k)}, x_2^{(k)}))_{k\in[3]}$. □

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

Bootle et al. [10] transformed their interactive protocol into a classical NIZK in the ROM using the Fiat-Shamir transform. Noticing that the two challenge sets $\mathbb{Z}_q$ and C have different size, they provided a more optimized soundness amplification technique. We explain in detail how we can incorporate such optimization technique when we instantiate the extractable LinHC protocol with the two constructions provided in Sec. 3.4. Since most of the argument is identical to those of the previous section, we refer the details to the full version.

5.3 Comparison

We compare Bootle et al's ROM secure NIZK and our QROM secure NIZK. We consider the application of proving knowledge of the ternary secret in LWE samples over $\mathbb{Z}_q$, which is commonly used in the literature to provide a basic benchmark, e.g., [10,7]. Such relation captures the setting of FHE schemes and group signatures. Aiming at the 128-bit quantum security level, our provably quantum secure NIZK has a proof size of 2071 KB while Bootle et al's (heuristically quantum secure) NIZK has proof size of 812 KB.[21] The overhead is around a factor of 2.6. The full detail on how we arrive at these values is provided in the full version. In contrast, if assume we were able to make Bootle et al's NIZK secure in the QROM using the extended Unruh transform [14] (see Footnote 4), the proof size becomes 44.9MB, where the overhead is a larger factor of 51.8. For completeness, we provide the details in the full version. Finally, note that it is unclear whether the Fiat-Shamir transform in the QROM can be securely applied to Bootle et al's NIZK.

[21] Bootle et al. [10] provides a proof size of 384 KB. Ours is around two times larger since we require $t = 8$, unlike $t = 4$, to achieve a minimal level of post-quantum security. Moreover, we do not reuse the commitment $t_{3,i}$ for all $i \in [t]$ as in [10] since it would harm zero-knowledge.

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. Due to page limitation, below we only remark on one of the recent lattice-based protocols. We provide further discussion in the full version for the rest of the protocols: proof of opening of commitments [3], one-out-of-many proofs [21], exact sound proofs for quadratic relations [48], and product proofs for commitments [1].

[21]: Range proofs. Range proof allows one to prove that a committed value resides in a specific range and is used in applications such as confidential transactions in cryptocurrencies. Recently, Esgin et al. [21] provided an efficient range proof by using new ideas on CRT-packing supporting "inter-slot" operations and NTT-friendly tools that permit the use of fully-splitting rings. It can be checked that the Σ-protocol for the range relation provided in [21, Theorem 1] is compatible with extractable LinHC protocols. Although it was not necessary for their scheme, we can modify the verifier in [21, 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 1.

Finally, we elucidate an inconvenient feature of some of the recent advanced lattice-based protocols. While conventional protocols only require 2 to 3 valid transcripts for special soundness, as much as 32 valid transcripts is required in the recent protocols [1]. Therefore, even if the protocols came with a compatible lossy function as in the definition of [35], the Fiat-Shamir transform incurs an extremely large reduction loss. Combining [18, Lemma 29] and [35, Theorem 1], a knowledge extractor (for the underlying protocol) given black-box access to a quantum adversary outputting a valid NIZK proof with probability ϵ after making Q hash queries, is only guaranteed in extracting a witness with probability $(\epsilon/Q^2)^{2\times 32-1} = \epsilon^{63}/Q^{126}$. In such cases, extractable LinHC protocols may 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

1. T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. *CRYPTO 2020*,.
2. C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. *CRYPTO 2018*.
3. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. *SCN 2018*.

4. M. Bellare. Lectures on nizks: A concrete security treatment. Lecture Notes.
5. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. *EUROCRYPT 2003*.
6. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. *ASIACRYPT 2014*.
7. W. Beullens. Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes. *EUROCRYPT 2020*.
8. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. *ASIACRYPT 2011*.
9. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. *CRYPTO 2013*.
10. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. *CRYPTO 2019*.
11. E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. *ACM CCS 2004*.
12. P. Chaidos and G. Couteau. Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. *EUROCRYPT 2018*.
13. M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. *ACM CCS 2017*.
14. M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe. SOFIA: $\mathcal{MQ}$-based signatures in the QROM. *PKC 2018*.
15. C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, and K. Xagawa. Modfalcon: compact signatures based on module ntru lattices. Cryptology ePrint Archive, Report 2019/1456.
16. M. Ciampi, G. Persiano, L. Siniscalchi, and I. Visconti. A transform for NIZK almost as efficient and general as the Fiat-Shamir transform without programmable random oracles. *TCC 2016*.
17. J. Don, S. Fehr, and C. Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. Cryptology ePrint Archive, Report 2020/282.
18. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. *CRYPTO 2019*.
19. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. *CRYPTO 2013*.
20. Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. *TCC 2006*.
21. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. *CRYPTO 2019*.
22. M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, and D. Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. *ACNS 2019*.
23. S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the Fiat-Shamir transform. *INDOCRYPT 2012*.
24. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *CRYPTO 1986*.
25. M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. *CRYPTO 2005*.

26. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *40th ACM STOC*.
27. J. Groth, R. Ostrovsky, and A. Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM (JACM)*, 2012.
28. A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. *PKC 2016*.
29. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. *ASIACRYPT 2008*.
30. D. Kales and G. Zaverucha. An attack on some signature schemes constructed from five-pass identification schemes. *CANS 2020*.
31. E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. *EUROCRYPT 2018*.
32. N. Koblitz and A. J. Menezes. Another look at "provable security". *Journal of Cryptology*, 2007.
33. Y. Lindell. An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. *TCC 2015*.
34. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 2015.
35. Q. Liu and M. Zhandry. Revisiting post-quantum Fiat-Shamir. *CRYPTO 2019*.
36. A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. *44th ACM STOC*.
37. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. *ASIACRYPT 2009*.
38. V. Lyubashevsky. Lattice signatures without trapdoors. *EUROCRYPT 2012*.
39. U. Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *Designs, Codes and Cryptography*, 2015.
40. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*.
41. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *37th ACM STOC*.
42. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 2009.
43. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. *EUROCRYPT 2018*.
44. J. Stern. A new identification scheme based on syndrome decoding. *CRYPTO'93*.
45. D. Unruh. Quantum proofs of knowledge. *EUROCRYPT 2012*.
46. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. *EUROCRYPT 2015*.
47. D. Unruh. Post-quantum security of Fiat-Shamir. *ASIACRYPT 2017*.
48. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. *CRYPTO 2019*.
49. M. Zhandry. How to construct quantum random functions. In *53rd FOCS*.
50. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. *CRYPTO 2012*.
51. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. *CRYPTO 2019*.