Quantum Rewinding for Many-Round Protocols

(Full Version)

Russell W. F. Lai[1]*, Giulio Malavolta[2]**, and Nicholas Spooner[3]***

[1] Aalto University, Finland
[2] Max Planck Institute for Security and Privacy, Germany
[3] University of Warwick, United Kingdom

Abstract. We investigate the security of succinct arguments against quantum adversaries. Our main result is a proof of knowledge-soundness in the post-quantum setting for a class of multi-round interactive protocols, including those based on the recursive folding technique of Bulletproofs.
To prove this result, we devise a new quantum rewinding strategy, the first that allows for rewinding across many rounds. This technique applies to any protocol satisfying natural multi-round generalizations of special soundness and collapsing. For our main result, we show that recent Bulletproofs-like protocols based on lattices satisfy these properties, and are hence sound against quantum adversaries.

Keywords: succinct arguments, lattice, Bulletproofs, quantum, knowledge-soundness

1 Introduction

Succinct arguments [Kil92,Mic94] allow a prover to convince a verifier that a statement x belongs to a language $\mathcal{L}$, with communication shorter than the witness length for the corresponding relation. Succinct arguments have become a cornerstone of modern cryptography and fueled the development of many real-world applications, such as verifiable computation and anonymous cryptocurrencies. Recent years have seen an explosion of new constructions of succinct arguments, based on a variety of cryptographic assumptions.

However, the advent of quantum computation poses a significant threat to these advancements. On the one hand, Shor's algorithm [Sho94] forces us to transition to cryptographic systems based on *post-quantum* assumptions, such as the hardness of the learning with errors (LWE) problem [Reg05]. On the other hand, some known techniques to prove security of cryptographic protocols no longer apply in the post-quantum regime, due to the fundamentally different nature of quantum information. Most notable are *rewinding* techniques, which are ubiquitous in security proofs for succinct arguments.

In a rewinding proof, it is argued that an adversary that succeeds on a single random challenge with high enough probability must succeed on multiple challenges. This classically intuitive idea fails in the quantum setting, because measuring the adversary's response to one challenge causes an irreversible loss of information which may render it useless for answering other challenges.

An important family of succinct arguments are interactive protocols based on the recursive folding technique of [BCC+16,BBB+18], also known in the literature as *Bulletproofs*. Leveraging algebraic properties of cryptographic schemes, Bulletproofs-like protocols can achieve much smaller proof sizes than PCP- and IOP-based succinct arguments [Kil92,BCS16] while retaining the benefit of a public-coin setup. Unlike PCP- and IOP-based arguments, however, the original Bulletproofs constructions are not post-quantum secure, being based on the hardness of the discrete logarithm problem. This has motivated a line of work that aims to design "post-quantum Bulletproofs" [BLNS20,AL21,ACK21,BCS21]. While these works do not rely on cryptographic assumptions which are quantum-insecure, their analysis of post-quantum security is only *heuristic*, in the sense that soundness is only shown against a *classical* adversary. Motivated by this state of affairs, we ask the following question:

* `russell.lai@aalto.fi`
** `giulio.malavolta@hotmail.it`
*** `nicholas.spooner@warwick.ac.uk`

Can we prove post-quantum security for Bulletproofs-like protocols?

Known techniques for rewinding quantum adversaries [Unr12,CMSZ21] do not appear to generalize to multi-round challenge-response protocols, let alone to logarithmic-round protocols like Bulletproofs. Thus, answering the above question requires us to develop new quantum rewinding techniques.

1.1 Our Results

In this work, we show that a class of "recursive" many-round interactive protocols is knowledge-sound against quantum adversaries. As a special case, we establish that lattice-based Bulletproofs protocols are post-quantum secure, assuming the quantum hardness of LWE. Loosely speaking, our main result can be restated as follows.

Theorem 1 (Informal version of Theorem 4). *Assuming the quantum hardness of the (Ring-)LWE problem, lattice-based Bulletproofs protocols are knowledge-sound against quantum algorithms.*

Our main result is obtained by developing two technical contributions of independent interest:

Fold-Collapsing Hash: We show that the lattice-based hash function $\mathsf{Hash}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$, where $\mathbf{A}$ is sampled uniformly at random and $\mathbf{x}$ is a "short" vector, satisfies a strong *collapsing* property[4]. Intuitively, we show that $\mathsf{Hash}_{\mathbf{A}}$ remains collapsing even when the key $\mathbf{A}$ is compressed via linear combinations of its columns with coefficients being short units in the base ring. This *fold-collapsing* property can be based on a variety of computational assumptions, including the (Ring-)LWE assumption.

Quantum Tree Rewinding: We develop a new quantum rewinding technique that allows us to extract from multi-round interactive protocols with certain collapsing and "recursive special soundness" properties. Our method combines the state-repair procedure of [CMSZ21] with a probability estimation step that determines the success probability of the adversary on a given sub-tree. Combined with the collapsing property above and the recursive special soundness of Bulletproofs-like arguments, this establishes the post-quantum security of these protocols.

1.2 Related Work

The witness folding technique for constructing succinct arguments was first introduced by Bootle et al. [BCC$^+$16] and later optimized by Bünz et al. [BBB$^+$18], who called their protocols Bulletproofs. The term "Bulletproofs" is now used to refer to a family of succinct arguments with a certain recursive structure. The early Bulletproofs protocols [BCC$^+$16,BBB$^+$18] prove quadratic relations of exponents of elements in prime-order cyclic groups, and their soundness relies on the discrete logarithm assumption over these groups. Lai, Malavolta, and Ronge [LMR19] generalized the folding technique to prove quadratic relations over bilinear pairing groups under a variant of the discrete logarithm assumption defined over these groups. As the discrete logarithm problems can be solved by Shor's algorithm [Sho94] in quantum polynomial time, none of these protocols are post-quantum sound.

While it is necessary to consider non-linear relations to obtain an argument for NP, Attema and Cramer [AC20] showed how to linearize the non-linear relations using secret-sharing techniques, and apply the folding technique to compress the argument for the linearized relations. Although their protocols for proving linear relations over groups are in fact *unconditionally* sound, they are trivial in the quantum setting because the relations that they prove are in BQP.

Bootle et al. [BLNS20] adapted the Bulletproofs folding technique to the lattice setting, giving a succinct argument for proving knowledge of the witness of a short integer solution (SIS) instance, i.e. a short vector $\mathbf{x}$ satisfying $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, over the m-th cyclotomic ring with m being a power of 2. The protocol, however, has large "slack": the knowledge extractor is only able to extract a short vector $\mathbf{x}'$ satisfying $\mathbf{A}\mathbf{x}' = 8^t \cdot \mathbf{y} \bmod q$, where $\ell = 2^t$ is the dimension of the witness $\mathbf{x}$. Albrecht and Lai [AL21] revisited this protocol and reduced

[4] Collapsing can be thought of as the quantum analogue of collision-resistance, and loosely speaking it requires that it is hard to determine whether a register containing valid pre-images of a given $\mathbf{y}$ was measured or not.

the slack from 8^t to 2^t with a careful choice of the challenge set R. They further eliminated the slack in the case of prime-power cyclotomic rings, i.e. when m is a power of a polynomially-large prime. Attema, Cramer, and Kohl [ACK21] improved the soundness analysis of [BLNS20,AL21], reducing the knowledge error from $O(\log \ell / |R|)$ to $2 \log \ell / |R|$, which is tight. Bootle, Chiesa, and Sotiraki [BCS21] proposed the abstract framework of sumcheck arguments which captures all Bulletproofs-like protocols, particularly lattice-based ones, mentioned above. Although lattice-based Bulletproofs for proving SIS relations are shown to be unconditionally sound against classical provers, the security proofs implicitly assume that the success probability of a prover remains unchanged after rewinding, which is generally false in the quantum setting.

1.3 Organization

In Section 2 we give an overview of our technical results. In Section 3 we recall standard preliminaries. In Section 4 we recall the notion of public-coin interactive arguments and introduce the notions of recursive special soundness and last-round collapsing. In Section 5 we show that protocols satisfying these properties are also knowledge-sound, even against quantum provers. In Section 6 we study the collapsing properties of hash function families implicit in lattice-based Bulletproof protocols. In Section 7 we build upon the results of Section 6 to show that lattice-based Bulletproof protocols are recursive special sound and last-round collapsing, and hence knowledge-sound, even against quantum provers.

2 Technical Overview

We give a brief overview of the main technical steps of our work. Before delving into the details of our analysis, we summarize the main conceptual steps of our proof:

Step I: We formalize a family of public-coin protocols Σ that satisfy two main properties of interest, namely recursive special soundness and last-round collapsing.

Step II: We describe a new quantum rewinding strategy that allows us to extract a witness from any recursive special sound and last-round collapsing protocol of the above defined family.

Step III: We show that the lattice-based hash function $\mathsf{Hash}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ is fold-collapsing, assuming that the (Ring-)LWE problem is intractable for quantum algorithms.

Step IV: Using the result from the previous step, we show that lattice-based Bulletproofs protocols are recursive special sound and last-round collapsing.

The remainder of the technical overview will be split into two parts, detailing Step I-II and Step III-IV respectively.

2.1 Quantum Rewinding

We first establish some context. Consider a $(2t+1)$-message public-coin interactive argument Σ where both the prover and the verifier input a statement x and the prover additionally inputs a witness w. The first $2t$ rounds of the protocol consists of the prover sending a "commitment" z_i and the verifier sending a challenge r_i for $i \in [t]$. The protocol ends with the prover sending a response w_{t+1} and the verifier outputting a single bit. The protocol Σ is k-tree-special sound, or $(k, \ldots, k)$-special sound, for a relation $\mathfrak{R}$ if the following holds: There exists an efficient extractor E which, given a statement x and complete k-ary tree of (edge-)depth t where the nodes and edges in each root-to-leaf path are labelled by a transcript $(z_1, r_1, \ldots, z_t, r_t, w_{t+1})$ of Σ which is accepting, extracts a witness w satisfying $\mathfrak{R}(x, w) = 1$.

In the following, we first review how tree-special soundness classically implies knowledge-soundness, and discuss where the classical reduction fails in the quantum setting. We then overview how post-quantum knowledge-soundness can be proven for protocols which satisfy a strengthening of tree special soundness along with a natural "collapsing" property.

Classical Tree Rewinding. To prove that a tree-special sound argument is knowledge-sound, the classical extraction proof (e.g. given in [BLNS20]) is based on the tree extraction technique of [BCC+16]. This technique obtains a k-ary tree of transcripts using a simple recursive strategy. This tree can then be provided to E in order to obtain the witness. For $i \in [t]$, [BCC+16] define subtree extractors T_i which, given a transcript prefix, obtain a k-ary subtree rooted at that prefix:

```
$T_i(r_1, \ldots, r_{i-1})$:
1. Let $\tau$ be a graph containing a single (root) node $v$.
2. Query the adversary at $(r_1, \ldots, r_{i-1})$ to obtain the $i$-th round commitment $z_i$. Label $v$ with $z_i$.
3. Repeat until $v$ has $k$ children: Choose $r_i \leftarrow R_i$ uniformly at random, and run $\tau' \leftarrow T_{i+1}(r_1, \ldots, r_i)$. If
   $T_{i+1}$ does not abort, attach $\tau'$ to $v$ via an edge labelled with $r_i$.
   If $T_{i+1}$ aborts, and this is the first loop iteration, then abort.
4. Return $\tau$.
```

The base case $T_{t+1}(r_1, \ldots, r_t)$ queries the adversary at $(r_1, \ldots, r_t)$ to obtain a full protocol transcript $(z_1, \ldots, z_{t+1})$ and returns z_{t+1} if that transcript is accepting (and otherwise aborts). The italicized condition above ensures that the procedure runs in expected polynomial time. Concretely, let ε denote the probability for r_i chosen uniformly at random that $T_{i+1}(r_1, \ldots, r_i)$ does not abort. The number of calls that T_i makes to T_{i+1} is then 1 with probability $1 - \varepsilon$ and $1 + (k-1)/\varepsilon$ (in expectation) with probability ε. Hence the overall expected number of calls is k, and by induction T_i runs in expected time $O(k^{t-i} \cdot t_A)$, where t_A is the running time of the adversary.

Quantum Tree Rewinding. Moving now to the quantum setting, the immediate problem is that Step 3 is a rewinding step: The above argument implicitly uses the fact that a classical adversary can be rewound to ensure that the success probability of T_{i+1} in each iteration is always ε. For quantum adversaries, the situation is more complicated, since measurements are in general irreversible operations. Known techniques [Unr12,CMSZ21] allow one to recover this type of rewinding in the quantum setting, provided the protocol satisfies a special "collapsing" condition.

Roughly speaking, this condition says the measurement performed by the reduction in the rewinding loop to obtain the response (in this case τ) is indistinguishable (to the adversary) from a binary measurement of whether the obtained response is valid or not (in this case, whether T_{i+1} aborts). Unfortunately, for the extractor above for general tree-special sound protocols we do not have this guarantee. The issue is that τ contains information about the set of challenges *to which the adversary produces an accepting response*. Measuring this information can cause the adversary's state to be disturbed in a detectable way. As a result, we do not know how to achieve general tree extraction in the quantum setting.

Instead, we observe that Bulletproofs-like protocols satisfy additional structural properties such that extracting the full tree is not necessary. Specifically, we can identify a family of protocols $(\Sigma_i)_{i=0}^{t}$ associated to Σ, where Σ_i has $2i+1$ messages, $\Sigma_t = \Sigma$ and Σ_0 is a noninteractive protocol where the prover sends w and the verifier checks $\mathfrak{R}(x, w)$.

This family has the property that, given a k-ary tree of accepting transcripts for Σ_i, we can obtain a k-ary tree of accepting transcripts for Σ_{i-1} by applying only *local* operations at the i-th layer: specifically, we compute a new label for each node v_i at depth i by applying a function E_i to the labels of its children. With this structural property, we can modify T_i (for all i) to directly output a witness (label) w_i instead of a tree τ. As a result, T_0 will directly output a witness w for x.

Moreover, we identify that if each Σ_i satisfies another property called *last-round collapsing* and T_{i+1} is executed *projectively* by T_i, then measuring the output of T_{i+1} is in fact indistinguishable from a binary measurement.

The Extractor. Assuming the protocol satisfies the recursive special soundness and last-round collapsing properties, which will be discussed later in this overview, it remains to show that T_{i+1} can indeed be executed projectively in a way that preserves the efficiency of the extractor as a whole. This turns out to be a significant challenge, as we now discuss.

We require an efficient unitary dilation of T_{i+1}, which in the classical setting is an expected polynomial time algorithm. Unfortunately, the unitary dilation of an expected (quantum) polynomial time (EQPT) algorithm is not generally efficiently implementable.[5] To avoid this problem, we design an extractor where the recursive call is to a *strict* polynomial-time algorithm. To give a sense of our construction, we will (for now) return to the classical setting. A natural first attempt is to simply *truncate* T_{i+1} to some strict number of repetitions N; applying this to all layers of the tree yields an extractor that makes N^t calls to the adversary. How large does N need to be? By Markov's inequality, the error incurred by truncation is $O(k/N)$; hence to achieve any guarantee, we require that $N = \Omega(k/\varepsilon)$. As a result, N^t is superpolynomial (since ε is an arbitrary inverse polynomial).

The key to overcoming this issue is to ensure that, no matter how many repetitions of Step 3 we execute, we only make k recursive calls. In particular, we must guarantee that whenever we make a call to T_{i+1}, it succeeds with high probability. To do this, we modify the extractor as follows.

```
$T_{i,\varepsilon}(r_1, \ldots, r_{i-1})$:
1. Repeat at most $N$ times until $|W| = k$:
   (a) Choose $r_i \leftarrow R_i$ uniformly at random.
   (b) Estimate $\varepsilon' \leftarrow \mathsf{Pr}_{r_{i+1},\ldots,r_t}[A(r_1, \ldots, r_t) \text{ convinces } V]$.
   (c) If $\varepsilon' \geq \varepsilon - \beta$, compute $w_i \leftarrow T_{i+1,\varepsilon-\beta}(r_1, \ldots, r_i)$. Add $(r_i, w_{i+1})$ to $W$.
2. Return $w_i \leftarrow E_i(W)$.
```

Note that we explicitly provide T with a lower bound ε on the success probability of A. We choose $\beta = 1/\mathsf{poly}(\lambda)$ to be small enough so that the adversary still has high enough success probability at the base of the recursion. The estimation step must be accurate to within an additive $o(\beta) = 1/\mathsf{poly}(\lambda)$ factor, which can be achieved using polynomially many calls to A. By Markov's inequality, the probability that $\varepsilon' < \varepsilon - \beta$ is at most β, and so by setting $N = O(\lambda/\beta) = \mathsf{poly}(\lambda)$ we see k successful iterations with probability $2^{-\lambda}$. Thus the overall size of $T_{i,\varepsilon}$ for inverse polynomial ε is The running time of $T_{i,\varepsilon}$ is then $k \cdot |T_{i+1,\varepsilon-\beta}| + N \cdot \mathsf{poly}(\lambda) = O(k^{t-i} \cdot \mathsf{poly}(\lambda))$.

Instantiating the above template in the quantum setting requires some care. The estimation step is achieved using e.g. the Marriott-Watrous algorithm [MW04] as described in [CMSZ21]. We facilitate the main rewinding loop using the state repair technique of [CMSZ21]. The state repair technique recovers the success probability of a state after it is disturbed by a (binary) projective measurement. In our setting, this measurement is "does the estimation step output $\varepsilon' \geq \varepsilon - \beta$?" All of these procedures have associated error; this error must be managed to ensure that it does not increase too much throughout the recursion. For more details, we refer the reader to Section 5.

2.2 Lattice-based Bulletproofs

In the above, we established that if a $(2t+1)$-message public-coin argument Σ induces a family $(\Sigma_i)_{i=0}^t$ which is *recursive special sound*, and each Σ_i is *last-round collapsing*, then Σ has post-quantum knowledge-soundness. In the following, we consider the case where Σ is a lattice-based Bulletproofs protocol, describe what it means for $(\Sigma_i)_{i=0}^t$ to be *recursive special sound* and Σ_i to be *last-round collapsing*, and outline how the properties can be achieved.

We recall the lattice-based Bulletproofs protocols from [BLNS20,AL21,ACK21]. In such protocols, both the prover and the verifier receive as input a SIS instance $(\mathbf{A}, \mathbf{y})$ defined over a ring $\mathcal{R}$,[6] and the prover additionally receives a short vector $\mathbf{x}$ satisfying $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$.[7] The interactive protocol consists of a

[5] [LMS21] proposes an extended computational model (in the context of zero knowledge simulation) which does permit this. However, this is not sufficient for our setting: While the model supports black-box access to unitary dilations of EQPT algorithms, here we would require a unitary dilation of an EQPT algorithm which itself calls the unitary dilation of an EQPT algorithm, etc.

[6] Rigorously, the matrix $\mathbf{A}$ is sampled uniformly at random by a setup algorithm, and is taken as input by the prover and the verifier as a public parameter.

[7] We focus only on the component of lattice-based Bulletproofs protocols where the witness folding technique is applied, since this is the technically challenging component in the quantum setting.

recursive application of a subroutine that allows the prover and the verifier to cut the size of the relation in half at each iteration: On input a hash key $\mathbf{A} = \mathbf{A}_0 \| \mathbf{A}_1$ and an image $\mathbf{y}$, the verifier samples a random (short) ring element r from a challenge set $R \subseteq \mathcal{R}$. The hash key is then "folded" by taking the appropriate linear combination of the columns $\mathbf{A}' = r \cdot \mathbf{A}_0 + \mathbf{A}_1$. Next, the prover updates the witness $\mathbf{x} = \mathbf{x}_0 \| \mathbf{x}_1$ to $\mathbf{x}' = \mathbf{x}_0 + r \cdot \mathbf{x}_1$, thus defining a new SIS instance $(\mathbf{A}', \mathbf{y}')$ satisfying

$$\begin{aligned}\mathbf{A}'\mathbf{x}' &= (r \cdot \mathbf{A}_0 + \mathbf{A}_1)(\mathbf{x}_0 + r \cdot \mathbf{x}_1)\\ &= \mathbf{A}_1\mathbf{x}_0 + r \cdot (\mathbf{A}_0\mathbf{x}_0 + \mathbf{A}_1\mathbf{x}_1) + r^2 \cdot \mathbf{A}_0\mathbf{x}_1\\ &= \underbrace{\mathbf{A}_1\mathbf{x}_0}_{\mathbf{l}} + r \cdot \mathbf{y} + r^2 \cdot \underbrace{\mathbf{A}_0\mathbf{x}_1}_{\mathbf{r}} = \mathbf{y}'\end{aligned}$$

where the terms $(\mathbf{l}, \mathbf{r})$ are sent by the prover to help the verifier compute the new image $\mathbf{y}'$. This effectively reduces the dimension of the statement by half. Repeating this procedure t-times, where $\ell = 2^t$ is the dimension of the witness $\mathbf{x}$, brings the dimension down to 1, at which point the prover can simply send the witness in the plain to the verifier.

Recursive Special Soundness. To define recursive special soundness, we first specify the family of protocols $(\Sigma_i)_{i=0}^t$ induced by a lattice-based Bulletproofs protocol Σ. For each i, the $(2i+1)$-message protocol Σ_i applies the folding technique recursively on the input statement $(\mathbf{A}, \mathbf{y})$ for i times, each taking 2 messages, and the final message is simply the witness $\mathbf{x}_i$ of the i-th folded statement $(\mathbf{A}_i, \mathbf{y}_i)$. Note that Σ_0 is the trivial 1-message protocol where the prover simply sends the witness $\mathbf{x}$ of $(\mathbf{A}, \mathbf{y})$, while $\Sigma_t = \Sigma$. Recursive special soundness requires that, for each $i \in [t]$, given k accepting transcripts (for Bulletproofs $k = 3$) for Σ_i that differ only in the last challenge-response rounds (i.e. messages $2i$ and $2i+1$), it is possible to efficiently recover a valid last-round (i.e. $(2i-1)^{\text{th}}$) message for the protocol Σ_{i-1}. From this definition, we can see that given a complete k-ary tree of accepting transcripts for Σ_t, it is possible to recursively recover a valid prover message $\mathbf{x}$ for the trivial protocol Σ_0.

With its close connection to the standard special soundness property, it is natural that the recursive special soundness of $(\Sigma_i)_{i=0}^t$ can be proven similarly: Given an accepting transcript of Σ_i of the form

$$(\mathbf{A}, \mathbf{y}, (\mathbf{l}_1, \mathbf{r}_1), r_1, \ldots, (\mathbf{l}_{i-1}, \mathbf{r}_{i-1}), r_{i-1}, (\mathbf{l}_i, \mathbf{r}_i), (r_i^{(j)}, \mathbf{x}_i^{(j)})_{j \in [k]})$$

the extractor E_i first derives $(\mathbf{A}_i, \mathbf{y}_i^{(j)})_{j \in [k]}$ satisfying

$$\mathbf{A}_i \begin{pmatrix} \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \mathbf{x}_i^{(3)} \end{pmatrix} = \mathbf{y}_i \bmod q,$$

then extracts $\mathbf{x}_{i-1}$ satisfying $\mathbf{A}_{i-1}\mathbf{x}_{i-1} = \mathbf{y}_{i-1} \bmod q$, provided that the challenges $(r_i^{(j)})_{j \in [k]}$ are chosen from a subtractive set [AL21][8]. The tuple

$$(\mathbf{A}, \mathbf{y}, (\mathbf{l}_1, \mathbf{r}_1), r_1, \ldots, (\mathbf{l}_{i-1}, \mathbf{r}_{i-1}), r_{i-1}, \mathbf{x}_{i-1})$$

is then an accepting transcript of Σ_{i-1}. As usual, two subtleties in the lattice setting are that the norm of the witness is slightly increased with each extraction step, and that the extracted witness may only be a preimage of $s \cdot \mathbf{y}_{i-1}$ for some short slack element $s \in \mathcal{R}$. These soundness gap issues can be handled by making an appropriate choice of (extraction relation) $\mathfrak{R}$, and choosing the challenge set and other parameters carefully.

Fold-Collapsing. Finally, we describe what it means for Σ_i to be last-round collapsing and how it is achieved. Last-round collapsing requires that, provided an *accepting* transcript of Σ_i where all messages but the last one are measured, it is computationally hard to tell whether the last message was also measured or not. In

[8] A subtractive set, also known as an exceptional sequence, is a set of ring elements such that the difference between any distinct members is invertible over the ring.

the procotol Σ_i induced above, the last message consists of a witness $\mathbf{x}_i$ of the statement $(\mathbf{A}_i, \mathbf{y}_i)$ defined by the previous rounds of interaction. Importantly, $(\mathbf{A}_i, \mathbf{y}_i)$ is fixed by the first $2i$ messages of the protocol. Thus, proving the above property is equivalent to establishing that the hash function

$$\mathsf{Hash}_{\mathbf{A}_i}(\mathbf{x}_i) = \mathbf{A}_i \mathbf{x}_i \bmod q$$

is *collapsing* for all $i \in \{0, \ldots, t\}$. It is known that such function satisfies the collapsing property, if the key $\mathbf{A}$ is uniformly chosen [LZ19,ACL+22]. However, recall that $\mathbf{A}_i$ is obtained by progressively folding the original key $\mathbf{A}$, so we need to show that the function remains collapsing even after we perform such operations over the hash key. We refer to this notion as *fold-collapsing.*

Our strategy to prove that the function is fold-collapsing proceeds in three steps: First, we appeal to the well-known fact that collapsing is implied by the stronger notion of somewhere statistically binding (SSB). Loosely speaking, SSB requires that the hash function has an alternative key generation mode, which is (i) computationally indistinguishable from the original mode, and that (ii) makes the hash statistically binding for a chosen position (say the j-th one) of the pre-image. Second, we show that the function $\mathsf{Hash}_{\mathbf{A}}$ is SSB. This is done by embedding ciphertexts of a linearly homomorphic encryption (with the appropriate ciphertext space) as the columns of the key $\mathbf{A}$. In the alternative mode, the key $\tilde{\mathbf{A}}_j$ consists of

$$\tilde{\mathbf{A}}_j = \left(\mathsf{Enc}(0) \ldots \mathsf{Enc}(0) \quad \underbrace{\mathsf{Enc}(1)}_{j\text{-th position}} \quad \mathsf{Enc}(0) \ldots \mathsf{Enc}(0) \right).$$

Since $\mathsf{Hash}_{\tilde{\mathbf{A}}_j}$ is a linear function, by the linearly-homomorphic property of the encryption scheme, we have $\tilde{\mathbf{A}}_j \mathbf{x} = \mathsf{Enc}(x_j) \bmod q$. Then, by the correctness of the encryption scheme, the hash function statistically binds the j-th coordinate of $\mathbf{x}$, as desired. Finally, to show that the folded key is still SSB, it suffices to observe that if the challenge set R consists of only units, i.e. $R \subseteq \mathcal{R}^\times$, then $r\mathbf{A}_0 + \mathbf{A}_1$ still preserves the invariant that exactly one ciphertext is not an encryption of 0 for any $r \in R$, again invoking the linear homomorphism of the encryption scheme. Thus, the folded key is still statistically binding on exactly one position of the input vector. Repeating this process recursively yields the desired statement.

Conveniently, for each of the subtractive sets R' suggested in [AL21] to be used as a challenge set, all but one element (i.e. 0) in R' are units in $\mathcal{R}$. Instantiating R with $R' \setminus \{0\}$ therefore meets all our requirements.

Remark 1. We stress that all of our results concern the protocol in the interactive setting. In particular, it should be noted that all lattice-based Bulletproofs protocols have at most inverse polynomial soundness, due to the fact that the challenge space is only polynomial size. While one can always reduce this to negligible by sequentially repeating the protocol, parallel repetition for super-constant round arguments is much less well-understood. In the classical setting, this was recently solved for tree special sound protocols in [AF21]; we leave open the problem of extending this to the quantum setting. Note that this required to establish that existing lattice-based Bulletproofs protocols can be made non-interactive in the QROM via Fiat-Shamir; importantly, sequential repetition does not suffice.

3 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter. We write $[n] \coloneqq \{1, 2, \ldots, n\}$ and $\mathbb{Z}_n \coloneqq \{0, 1, \ldots, n-1\}$ for $n \in \mathbb{N}$. We write $\varphi(n)$ for the Euler totient function, i.e. the number of positive integers at most and coprime with n. If a is a ring element, we write $\langle a \rangle$ for the ideal generated by a.

We make use of the following simple fact, a consequence of Markov's inequality.

Proposition 1. *Let X be a random variable supported on $[0, 1]$. Then for all $\alpha \geq 0$, $\Pr[X \geq \alpha] \geq E[X] - \alpha$.*

3.1 Lattices

For $m \in \mathbb{N}$, let $\zeta = \zeta_m \in \mathbb{C}$ be any fixed primitive m-th root of unity. We write $\mathbb{K} = \mathbb{Q}(\zeta)$ for the cyclotomic field of order $m \geq 2$ and degree $\varphi(m)$, and $\mathcal{R} = \mathbb{Z}[\zeta]$ for its ring of integers, called a cyclotomic ring for short. It is well-known that $\mathcal{R} \cong \mathbb{Z}[x]/\left\langle \Phi_m(x) \right\rangle$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial. For $q \in \mathbb{N}$, write $\mathcal{R}_q := \mathcal{R}/q \cdot \mathcal{R}$.

For elements $x \in \mathcal{R}$ we denote the infinity norm of its coefficient vector (with the powerful basis $\left\{1, \zeta, \ldots, \zeta^{\varphi(m)-1}\right\}$) as $\|x\|$. If $\mathbf{x} \in \mathcal{R}^k$ we write $\|\mathbf{x}\|$ for the infinity norm of $\mathbf{x}$.

The ring expansion factor of $\mathcal{R}$ is defined as $\gamma_{\mathcal{R}} := \max_{a,b \in \mathcal{R}} \frac{\|a \cdot b\|}{\|a\| \cdot \|b\|}$. By definition, we have for any $x, y \in \mathcal{R}$ that $\|x \cdot y\| \leq \gamma_{\mathcal{R}} \cdot \|x\| \cdot \|y\|$.

For any ordered set $T = (r_i)_{i \in \mathbb{Z}_t} \subseteq \mathcal{R}$, we write

$$\mathbf{V}_T := \begin{pmatrix} 1 & 1 & \ldots & 1 \\ r_0 & r_1 & \ldots & r_{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_0^{t-1} & r_1^{t-1} & \ldots & r_{t-1}^{t-1} \end{pmatrix}$$

for the (column-style) Vandermonde matrix induced by T.

Definition 1 ((s,t)-Subtractive Sets [AL21]). *Let $s \in \mathcal{R}$ and $t \in [n]$. A set $R \subseteq \mathcal{R}$ is said to be (s,t)-subtractive if for any t-subset $T = \{r_i\}_{i \in \mathbb{Z}_t} \subseteq R$, it holds that $s \in \langle \det(\mathbf{V}_T) \rangle$. If R is $(1,2)$-subtractive, we simply say that R is subtractive.*

Proposition 2 ([AL21]). *If m is a power of a prime p and $\mathcal{R}$ is the m-th order cyclotomic ring, then the set $R := \left\{1, 1+\zeta, \ldots, \sum_{i \in \mathbb{Z}_{p-1}} \zeta^i\right\} \subseteq_{p-1} \mathcal{R}$ is subtractive. Furthermore, for any ordered set $T = (r_0, r_1, r_2) \subseteq R$ and any $x_0, x_1, x_2 \in R$ with $\|x_j\| \leq \beta$,*

$$\left\| \begin{pmatrix} r_0 \cdot x_0 & r_1 \cdot x_1 & r_2 \cdot x_2 \\ x_0 & x_1 & x_2 \end{pmatrix} \cdot \mathbf{V}_T^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\| \leq 24 \cdot \varphi(m) \cdot \gamma_{\mathcal{R}} \cdot \beta.$$

If m is a power of 2 and $\mathcal{R}$ is the m-th order cyclotomic ring, then the set $R := \left\{1, \zeta, \ldots, \zeta^{\varphi(m)-1}\right\} \subseteq_{\varphi(m)} \mathcal{R}$ is $(2,3)$-subtractive. Furthermore, for any ordered set $T = (r_0, r_1, r_2) \subseteq R$ and any $x_0, x_1, x_2 \in R$ with $\|x_j\| \leq \beta$,

$$\left\| \begin{pmatrix} r_0 \cdot x_0 & r_1 \cdot x_1 & r_2 \cdot x_2 \\ x_0 & x_1 & x_2 \end{pmatrix} \cdot s \cdot \mathbf{V}_T^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\| \leq 3 \cdot \varphi(m) \cdot \gamma_{\mathcal{R}} \cdot \beta.$$

3.2 Quantum Information

We recall the basics of quantum information. Most of the following is taken almost in verbatim from [CMSZ21]. A (pure) *quantum state* is a vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$ with $\||\psi\rangle\| = 1$; in this work, $\mathcal{H}$ is finite-dimensional. We denote by $\mathbf{S}(\mathcal{H})$ the space of Hermitian operators on $\mathcal{H}$. A *density matrix* is a positive semi-definite operator $\rho \in \mathbf{S}(\mathcal{H})$ with $\mathrm{Tr}(\rho) = 1$. A density matrix represents a probabilistic mixture of pure states (a mixed state); the density matrix corresponding to the pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi|$. Typically we divide a Hilbert space into *registers*, e.g. $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$. We sometimes write, e.g., $\rho^{\mathcal{H}_1}$ to specify that $\rho \in \mathbf{S}(\mathcal{H}_1)$.

A unitary operation is a complex square matrix U such that $UU^\dagger = \mathbf{I}$. The operation U transforms the pure state $|\psi\rangle$ to the pure state $U|\psi\rangle$, and the density matrix ρ to the density matrix $U\rho U^\dagger$. We write $U(\mathcal{H})$ for the set of unitary operators on $\mathcal{H}$.

A *projector* Π is a Hermitian operator ($\Pi^\dagger = \Pi$) such that $\Pi^2 = \Pi$. A *projective measurement* is a collection of projectors $\mathsf{P} = (\Pi_i)_{i \in S}$ such that $\sum_{i \in S} \Pi_i = \mathbf{I}$. This implies that $\Pi_i \Pi_j = 0$ for distinct i and j in S. The application of P to a pure state $|\psi\rangle$ yields outcome $i \in S$ with probability $p_i = \|\Pi_i |\psi\rangle\|^2$; in this case the post-measurement state is $|\psi_i\rangle = \Pi_i |\psi\rangle / \sqrt{p_i}$. We refer to the post-measurement state $\Pi_i |\psi\rangle / \sqrt{p_i}$ as the result of applying P to $|\psi\rangle$ and *post-selecting* (conditioning) on outcome i. A state $|\psi\rangle$ is an *eigenstate* of P if it is an eigenstate of every Π_i. A two-outcome projective measurement is called a *binary projective measurement*, and is written as $\mathsf{P} = (\Pi, \mathbf{I} - \Pi)$, where Π is associated with the outcome 1, and $\mathbf{I} - \Pi$ with the outcome 0.

General (non-unitary) evolution of a quantum state can be represented via a *completely-positive trace-preserving (CPTP)* map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H}')$. We omit the precise definition of these maps in this work; we only use the facts that they are trace-preserving (for every $\rho \in \mathbf{S}(\mathcal{H})$ it holds that $\mathrm{Tr}(T(\rho)) = \mathrm{Tr}(\rho)$) and linear. For every CPTP map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H})$ there exists a *unitary dilation* U that operates on an expanded Hilbert space $\mathcal{H} \otimes \mathcal{K}$, so that $T(\rho) = \mathrm{Tr}_{\mathcal{K}}(U(\rho \otimes |0\rangle\langle 0|^{\mathcal{K}})U^\dagger)$. This is not necessarily unique; however, if T is described as a circuit then there is a dilation U_T represented by a circuit of size $O(|T|)$.

For Hilbert spaces $\mathcal{A}, \mathcal{B}$ the *partial trace* over $\mathcal{B}$ is the unique CPTP map $\mathrm{Tr}_{\mathcal{B}} \colon \mathbf{S}(\mathcal{A} \otimes \mathcal{B}) \to \mathbf{S}(\mathcal{A})$ such that $\mathrm{Tr}_{\mathcal{B}}(\rho_A \otimes \rho_B) = \mathrm{Tr}(\rho_B)\rho_A$ for every $\rho_A \in \mathbf{S}(\mathcal{A})$ and $\rho_B \in \mathbf{S}(\mathcal{B})$.

A *general measurement* is a CPTP map $\mathsf{M} \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H} \otimes \mathcal{O})$, where $\mathcal{O}$ is an ancilla register holding a classical outcome. Specifically, given measurement operators $\{M_i\}_{i=1}^N$ such that $\sum_{i=1}^N M_i M_i^\dagger = \mathbf{I}$ and a basis $\{|i\rangle\}_{i=1}^N$ for $\mathcal{O}$, $\mathsf{M}(\rho) = \sum_{i=1}^N (M_i \rho M_i^\dagger \otimes |i\rangle\langle i|^{\mathcal{O}})$. We sometimes implicitly discard the outcome register. A projective measurement is a general measurement where the M_i are projectors. A measurement induces a probability distribution over its outcomes given by $\Pr[i] = \mathrm{Tr}\Big(|i\rangle\langle i|^{\mathcal{O}} \, \mathsf{M}(\rho)\Big)$; we denote sampling from this distribution by $i \leftarrow \mathsf{M}(\rho)$. The *trace distance* between states ρ, σ, denoted $d(\rho, \sigma)$, is defined as

$$d(\rho, \sigma) = \frac{1}{2} \mathrm{Tr}\Big(\sqrt{(\rho - \sigma)^2}\Big).$$

The trace distance is contractive under CPTP maps (for any CPTP map T, $d(T(\rho), T(\sigma)) \leq d(\rho, \sigma)$). It follows that for any measurement M, the statistical distance between the distributions $\mathsf{M}(\rho)$ and $\mathsf{M}(\sigma)$ is bounded by $d(\rho, \sigma)$.

We also define a notion of quantum *computational* distinguishability. Specifically, for states ρ, σ,

$$d_{\mathsf{comp}}(\rho, \sigma)_N \coloneqq \max_{D, |D| \leq N} |\Pr[D(\rho) \to 1] - \Pr[D(\sigma) \to 1]| \ ,$$

where D is a quantum circuit. For sequences of states $(\rho_\lambda)_\lambda, (\sigma_\lambda)_\lambda$ we say that $d_{\mathsf{comp}}(\rho_\lambda, \sigma_\lambda) \leq \varepsilon + \mathsf{negl}(\lambda)$ if for all polynomials p, $d_{\mathsf{comp}}(\rho_\lambda, \sigma_\lambda)_{p(\lambda)} \leq \varepsilon + \mathsf{negl}(\lambda)$.

Clearly d_{comp} satisfies the triangle inequality and for all $\lambda \in \mathbb{N}$, $d_{\mathsf{comp}}(\rho, \sigma)(\lambda) \leq d(\rho, \sigma)$. For bipartite states on $\mathcal{A} \otimes \mathcal{B}$ we affix a superscript $\mathcal{A}$ to d and d_{comp} to indicate that the distance is with respect to $\mathcal{A}$ only, i.e.

$$d^{\mathcal{A}}(\rho, \sigma) = d(\mathrm{Tr}_{\mathcal{B}}(\rho), \mathrm{Tr}_{\mathcal{B}}(\sigma)) \ .$$

Gentle Measurement. We have the following *gentle measurement lemma*, which bounds how much a state is disturbed by applying a measurement whose outcome is almost certain.

Lemma 1 (Gentle Measurement [Win99]). *Let $\rho \in \mathbf{S}(\mathcal{H})$ and $\mathsf{P} = (\Pi, \mathbf{I} - \Pi)$ be a binary projective measurement on $\mathcal{H}$ such that $\mathrm{Tr}(\Pi\rho) \geq 1 - \delta$. Let*

$$\rho' = \frac{\Pi\rho\Pi}{\mathrm{Tr}(\Pi\rho)} \quad \text{and} \quad \rho'' = \Pi\rho\Pi + (I - \Pi)\rho(I - \Pi).$$

Then

$$d(\rho, \rho') \leq 2\sqrt{\delta} \quad \text{and} \quad d(\rho, \rho'') \leq 2\sqrt{\delta}.$$

Quantum Algorithms. In this work, a *quantum adversary* is a family of quantum circuits $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ represented classically using some standard universal gate set. A quantum adversary is *polynomial-size* if there exists a polynomial p and $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$ it holds that $|A_\lambda| \leq p(\lambda)$ (i.e., quantum adversaries have classical non-uniform advice).

A circuit C with *black-box* access to a unitary U, denoted C^U, is a standard quantum circuit with special gates that act as U and $U^\dagger$. We also use C^T to denote black-box access to a map T, which we interpret as C^{U_T} for a unitary dilation U_T of T; all of our results are independent of the choice of dilation. This allows, for example, the "partial application" of a projective measurement, and the implementation of a general measurement via a projective measurement on a larger space.

Interactive Quantum Circuits. We introduce the definition for interactive quantum circuits.

Definition 2. *A t-round interactive quantum circuit A is a sequence of maps $(U_1, \dots, U_t)$ where $U_i \colon R_i \to \mathcal{U}(\mathcal{I} \otimes \mathcal{Z}_i)$. We also denote by U_i the unitary $\sum_{r_i \in R_i} |r_i\rangle\langle r_i| \otimes U_i(r_i)$. The size of an interactive quantum circuit is the sum of the sizes of the circuits implementing the unitaries $U_1, \dots, U_t$.*

Let $P^* = (U_1, \dots, U_t, |\psi\rangle)$; then E^{P^*} is a quantum circuit with special gates corresponding to the unitaries U_i and $(U_i)^\dagger$ for $i \in [t]$. The requirement that the U_i be unitary is without loss of generality, in the sense that any interactive quantum adversary not of this form can be "purified" into a circuit of this form that is only a constant factor larger with the same observable behavior. Using this formulation, we can sample the random variable $\langle P^*(|\psi\rangle), V\rangle$ equivalently as:

1. Initialize the register $\mathcal{I}$ to $|\psi\rangle$, and $\tau = ()$.
2. For $i = 1 \dots t$:
   (a) Sample $r_i \leftarrow R_i$.
   (b) Apply unitary $U_i(r_i)$ to $\mathcal{I} \otimes \mathcal{Z}_i$.
   (c) Measure $\mathcal{Z}_i$ in the computational basis to obtain response z_i. Append (r_i, z_i) to τ.
3. Return the output of $V(\tau)$.

In particular, the interaction is *public coin*. Note again that we restrict the operation of P^* in each round to be unitary except for the measurement of $\mathcal{Z}_i$ in the computational basis.

4 Recursive Special Sound and Last-Round Collapsing Arguments

We recall the definitions of interactive arguments and their knowledge soundness. We then define the new notions of recursive special soundness and last-round collapsing.

Definition 3 (Arguments). *Let $i \geq 0$ be an integer. A $(2i+1)$-message public-coin argument system $\Pi = (\mathsf{Setup}, \Sigma = (P, V))$ consists of a PPT algorithm Setup and a $(2i+1)$-message protocol $\Sigma = (P, V)$ between an interactive PPT prover P and an interactive PPT verifier V, is associated to a tuple of spaces $(X, W, (Z_j, R_j)_{j\in[i]}, W_{i+1})$, and has the following structural properties:*

- *The Setup algorithm takes as input the security parameter 1^λ and outputs some public parameters pp.*
- *Both P and V receive as input the public parameters pp and a statement $x \in X$. The prover P additionally receives a witness $w \in W$.*
- *The public parameters, the statement x, and the $2i+1$ messages sent by P and V in the protocol Σ, called collectively a transcript, is labelled by $(\mathsf{pp}, x, z_1, r_1, \dots, z_i, r_i, w_{i+1})$, where $z_j \in Z_j$ sent by P are called commitments, $r_j \in R_j$ sent by V are called challenges, and $w_{i+1} \in W_{i+1}$ sent by P is called a response.*
- *The challenges r_j are sampled by V uniformly randomly from R_j.*[9]

[9] In general, r_j could be sampled from a public distribution over R_j.

A transcript $(\mathsf{pp}, x, z_1, r_1, \ldots, z_i, r_i, w_{i+1})$ *is said to be accepting for* Σ *if* $V(\mathsf{pp}, x, z_1, r_1, \ldots, z_i, r_i, w_{i+1}) = 1$. *A* k*-branch of transcripts of* Σ *is a tuple consisting of some public parameters, a statement, and a prefix of messages*

$$(\mathsf{pp}, x, z_1, r_1, \ldots, z_{i-1}, r_{i-1}, z_i)$$

along with k *distinct* i*-th round challenges* $(r_i^{(j)})_{j\in[k]}$*, and* k *responses* $(w_{i+1}^{(j)})_{j\in[k]}$. *A* k*-branch of transcripts is said to be accepting for* Σ *if*

$$(\mathsf{pp}, x, z_1, r_1, \ldots, z_{i-1}, r_{i-1}, z_i, r_i^{(j)}, w_{i+1}^{(j)})$$

is accepting for Σ *for all* $j \in [k]$.

Note that if $i = 0$ then the protocol is non-interactive: the transcript consists only of (pp, x, w_1).

For the protocols we consider, the statement to be proved depends on the public parameters pp. As such, we will define proofs of knowledge with respect to relations on triples (pp, x, w). Observe, in particular, that when $i = 0$ in Definition 3 the verifier itself defines such a relation. Our proof of knowledge definition is somewhat weaker than standard definitions of proof of knowledge in that the extractor is permitted a given additive inverse polynomial loss.

Definition 4 (Proof of knowledge). *We say that an argument system* $\Pi = (\mathsf{Setup}, \Sigma = (P, V))$ *is a (post-quantum) proof of knowledge with knowledge error* κ *for a relation* $\mathfrak{R}$ *if there exists a (quantum) polynomial-time extractor* E *and such that for any inverse polynomial* ν *and any (quantum) polynomial-size adversary* P^*,

$$\mathsf{Pr}\left[\mathfrak{R}(\mathsf{pp}, x, w) \;\middle|\; \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ w \leftarrow \mathsf{Extract}^{P^*}(\mathsf{pp}, x, 1^{1/\nu}) \end{array}\right] \geq \mathsf{Pr}\left[\langle P^*, V\rangle = 1\right] - \kappa(\lambda) - \nu(\lambda) \ .$$

Definition 5 (Recursive k-Special Soundness). *Let* $(\Pi_i)_{i=0}^t$ *be a family where* $\Pi_i = (\mathsf{Setup}, \Sigma_i = (P_i, V_i))$ *is a* $(2i+1)$*-message public-coin argument system with a common* Setup *algorithm associated to the spaces* $(X, W, (Z_j, R_j)_{j\in[i]}, W_{i+1})$. *The family* $(\Pi_i)_{i=0}^t$ *is said to be recursive* k*-special sound if for each* $i \in [t]$ *there exists an efficient extractor* E_i *satisfying the following properties:*

- *The extractor* E_i *takes as input* $(r_i^{(j)}, w_{i+1}^{(j)})_{j\in[k]} \in (R_i \times W_{i+1})^k$ *and outputs* $w_i \in W_i$.
- *If*

  $$(\mathsf{pp}, x, z_1, r_1, \ldots, z_{i-1}, r_{i-1}, z_i, (r_i^{(j)}, w_i^{(j)})_{j\in[k]})$$

  is an accepting k*-branch of transcripts for* Σ_i*, and* $w_i = E_i((r_i^{(j)}, w_{i+1}^{(j)})_{j\in[k]})$*, then*

  $$(\mathsf{pp}, x, z_1, r_1, \ldots, z_{i-1}, r_{i-1}, w_i)$$

  is an accepting transcript for Σ_{i-1}.

Definition 6 (Last-Round Collapsing). *Let* Π *be a* $(2i+1)$*-message public-coin argument system associated to the spaces* $(X, W, (Z_j, R_j)_{j\in[i]}, W_{i+1})$. *We say that* Π *is last round collapsing if for any efficient (quantum) adversary* A

$$\left|\Pr\left[\mathsf{LastRoundCollapsing}^0_{\Pi,A}(1^\lambda) = 1\right] - \Pr\left[\mathsf{LastRoundCollapsing}^1_{\Pi,A}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{LastRoundCollapsing}^b_{\Pi,A}$ *is defined as follows:*

$\underline{\mathsf{LastRoundCollapsing}^b_{\Pi,A}(1^\lambda)}$*:*

1. *The challenger generates* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. *The challenger runs* $x \leftarrow A(\mathsf{pp})$.

3. *The challenger executes the interaction $(A, V(\mathsf{pp}, x))$ up until measuring the last message of the adversary. Let $\tau = (\mathsf{pp}, x, z_1, r_1, \ldots, z_t, r_t)$ be the protocol transcript thus far (excluding the last message) and let $\mathcal{W}$ be the register that contains the state corresponding to the last message of the adversary.*
4. *Let V_τ be the unitary that acts on $\mathcal{W}$ and a fresh ancilla, and CNOTs into the fresh ancilla the bit that determines whether the transcript is valid. Apply V_τ, measure the ancilla, and apply $V_\tau^\dagger$.*
5. *If the output of the measurement is 0, then abort the experiment. Else proceed.*
6. *If $b = 0$ do nothing.*
7. *If $b = 1$ measure the register $\mathcal{W}$ in the computational basis, discard the result.*
8. *Return to A all registers and output whichever bit A outputs.*

5 Quantum Tree-Extraction

In this section we give an algorithm for extracting a witness from a recursively k-special sound, last-round collapsing protocol. We prove the following general theorem.

Theorem 2. *Let $(\Pi_i = (\mathsf{Setup}, \Sigma_i = (P_i, V_i)))_{i=0}^{t}$ be a recursively k-special sound family where Π_i is last-round collapsing for all i. Then Π_t is a post-quantum proof of knowledge for (the relation induced by) V_0 with knowledge error*

$$\sum_{i=1}^{t} \frac{k-1}{|R_i|} \ .$$

In Section 5.1 we give some notation which will be used in this section, and specify the quantum algorithms we require. We also prove a new result about the Repair algorithm of [CMSZ21], which gives a better characterization of the distribution of outcomes from repeated applications of the repair experiment; this is necessary for our main result. In Section 5.2 we specify our extractor and show that it runs in polynomial time. In Section 5.3 we prove that the extractor is correct.

5.1 Notation and quantum algorithms

For a classical predicate $f \colon R \times Z \to \{0,1\}$, let $\Pi_{f(r,\cdot)} := \sum_{z \in Z, f(r,z)=1} |z\rangle\langle z|_{\mathcal{Z}}$. Given also a mapping $U \colon r \to U(\mathcal{A}, \mathcal{Z})$, we define the Hermitian matrix $E_{U,f} := \frac{1}{|R|} \sum_{r \in R} U(r)^\dagger \Pi_{f(r,\cdot)} U(r)$. Let $\mathsf{T}_{\geq p}^{U,f} := (\Pi_{\geq p}^{U,f}, I - \Pi_{\geq p}^{U,f})$, where

$$\Pi_{\geq p}^{U,f} := \sum_{j, p_j \geq p} |j\rangle\langle j| \ ,$$

for $\sum_j p_j \, |j\rangle\langle j|$ the spectral decomposition of $E_{U,f}$. Note that $0 \leq p_j \leq 1$ for all j.

Lemma 2 ([Zha20,CMSZ21]). *There is a quantum algorithm $\mathsf{Estimate}_{\varepsilon,\delta}$ with the following guarantees. For any classical predicate $f \colon R \times Z \to \{0,1\}$, mapping $U \colon r \to U(\mathcal{A}, \mathcal{Z})$ and state $\rho \in \mathcal{A} \otimes \mathcal{Z}$:*

- $\mathbb{E}[\mathsf{Estimate}_{\varepsilon,\delta}^{U,f}(\rho)] = \mathrm{Tr}(E_{U,f}\rho) = \frac{1}{|R|} \sum_{r \in R} \mathrm{Tr}\big(\Pi_{f(r,\cdot)} U_r \rho U_r^\dagger\big)$*;*
- $\mathsf{Estimate}_{\varepsilon,\delta}^{U,f}$ *is (ε,δ)-almost projective; and*
- *For any $q \in [0,1]$,* $\mathsf{Estimate}_{\varepsilon,\delta}^{U,f}$*,*

$$\mathsf{Pr}\left[p \geq q \wedge b = 0 \,\middle|\, \begin{array}{r} (p, \rho') \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U,f}(\rho) \\ b \leftarrow \mathsf{T}_{\geq q-\varepsilon}^{U,f}(\rho') \end{array} \right] \leq \delta \ .$$

$\mathsf{Estimate}_{\varepsilon,\delta}^{U,f}$ *has quantum circuit complexity $O(|f| \cdot \frac{1}{\varepsilon} \log \frac{1}{\delta})$ given oracle access to $U := \sum_{r \in R} |r\rangle\langle r| \otimes U(r)$.*

We denote by $\mathsf{Threshold}_{\gamma,\varepsilon,\delta}$ the quantum algorithm which runs $\mathsf{Estimate}$ and outputs 1 if its output is at least γ and 0 otherwise.

We recall the state repair theorem of [CMSZ21].

```
$\mathsf{RepExpt}_T^{\mathsf{M},\mathsf{P}}$:
 1. (Estimate) Apply the $(\varepsilon,\delta)$-almost-projective measurement $\mathsf{M}$, obtaining outcome $p$;
 2. (Disturb) Apply the projective measurement $\mathsf{P}$, obtaining outcome $k \in [N]$;
 3. (Repair) Run $\mathsf{Repair}_T[\mathsf{M},\mathsf{P}](k,p)$.
 4. (Re-estimate) Apply $\mathsf{M}$ again, obtaining outcome $p'$.
 5. Output $(p,p')$.

$\mathsf{MultiExpt}_T^{(D_s)_{s=1}^N,(\mathsf{M}_i)_{i\in\mathbb{N}}}$:
 1. Apply $\mathsf{Estimate}$, obtaining outcome $p_0$.
 2. For $s = 1,\ldots,N$:
    (a) Apply $\mathsf{Estimate}$, obtaining outcome $p_s$.
    (b) Sample $i_s \leftarrow D_s(i_1,b_1,\ldots,i_{s-1},b_{s-1})$, and measure $b_s \leftarrow \mathsf{M}_{i_s}$.
    (c) Run $\mathsf{Repair}_T[\mathsf{Estimate},\mathsf{M}_{i_s}](b_s,p_s)$.
 3. Output $\sum_{s=1}^N b_s$.
```

Fig. 1. Experiments involving the Repair algorithm.

Theorem 3 (State repair, [CMSZ21]). *Let M be an (ε,δ)-almost projective measurement on $\mathcal{H}$, let P be an n-outcome projective measurement on $\mathcal{H}$, and let T be any positive integer. There is quantum procedure Repair such that $\mathsf{RepExpt}_T^{\mathsf{M},\mathsf{P}}$ (see Fig. 1) satisfies the following guarantee. For any state ρ on $\mathcal{H}$ and $(p,p') \leftarrow \mathsf{RepExpt}_T^{\mathsf{M},\mathsf{P}}(\rho)$:*

$$\Pr\left[|p'-p| > 2\varepsilon\right] \leq n(\delta + 1/T) + 4\sqrt{\delta}.$$

Moreover, Repair has quantum circuit complexity $O(T)$ given oracle access to P and $\overline{\mathsf{M}}$.

We prove the following lemma, which shows that the outcomes b_s in $\mathsf{MultiExpt}$ are essentially "independent" of the sequence of prior outcomes.

Lemma 3. *For each $s = 1,\ldots,N$, let D_s be a randomized function that takes an element of $(\mathbb{N}\times\{0,1\})^{s-1}$ and outputs $i_s \in \mathbb{N}$. Let $(\mathsf{M}_i)_{i\in\mathbb{N}}$ be a list of measurements.*

For any state $\rho \in \mathbf{S}(\mathcal{A}\otimes\mathcal{Z})$, the following holds:

$$\Pr_{S\leftarrow\mathsf{MultiExpt}(\rho)}\left[S < k\right] \leq \Pr\left[\sum_{i=1}^N Y_s < k\right] + N/T + O(\sqrt{\delta} + (N/T)^2)\ ,$$

where the $(Y_s)_{s=1}^N$ are distributed as follows:

1. *Apply $\mathsf{Estimate}_{\varepsilon,\delta}^{U,f}$ to ρ, obtaining outcome p_0. Let $\alpha := p_0 - 2\varepsilon N$.*
2. *For each $s \in [N]$, sample Y_s from a Bernoulli distribution with parameter*

$$\zeta := \min_{|v\rangle\in\mathrm{im}(\Pi_{\geq\alpha}^{U,f})}\ \min_{\substack{\mathbf{i}\in\mathbb{N}^{s-1}\\ \mathbf{b}\in\{0,1\}^{s-1}}} \mathbb{E}_{i_s\leftarrow D_s(\mathbf{i},\mathbf{b})}\Pr\left[\mathsf{M}_{i_s}(|v\rangle\langle v|) \to 1\right]\ .$$

Proof. By Theorem 3, for each $s \in [N]$ it holds that

$$\Pr\left[p_s < p_{s-1} - 2\varepsilon\right] \leq 2/T + O(\sqrt{\delta})\ .$$

Denote by E the event that, for any $s \in [N]$, $p_s < p_{s-1} - 2\varepsilon$. By a union bound,

$$\Pr\left[E\right] \leq 2N/T + O(N\sqrt{\delta} + (N/T)^2)\ .$$

Now consider the following hybrid experiment:

```
$\mathsf{Hyb}^{\mathsf{P}}_{T}$:
  1. Apply $\mathsf{Estimate}$, obtaining outcome $p_0$.
  2. For $s = 1, \dots, N$:
     (a) Apply $\mathsf{Estimate}$, obtaining outcome $p_s$.
     (b) Apply $\mathsf{T}^{U,f}_{\geq p_s - \varepsilon}$, and *postselect* on obtaining outcome 1.
     (c) Sample $(i_s, \mathsf{M}) \leftarrow D_s(i_1, b_1, \dots, i_{s-1}, b_{s-1})$, and measure $b_s \leftarrow \mathsf{M}$.
     (d) Run $\mathsf{Repair}_T[\mathsf{Estimate}, \mathsf{M}_i](b_s, p_s)$.
```

By Lemma 2, in each iteration s, $\mathsf{T}^{U,f}_{\geq p_s - \varepsilon}$ yields outcome 1 with probability at least $1 - \delta$. Hence by gentle measurement, $d(\mathsf{MultiExpt}, \mathsf{Hyb}) = O(N\sqrt{\delta})$. Switching to Hyb, it holds by definition of ζ that

$$\mathsf{Pr}_{\mathsf{Hyb}}\left[b_s = 1 \mid \neg E, i_1, b_1, \dots, i_{s-1}, b_{s-1}\right] \geq \zeta \ .$$

Therefore the distribution of $(\sum_{s=1}^{N} b_s \mid \neg E)$ induced by Hyb stochastically dominates $\sum_{s=1}^{N} Y_s$; that is, for all k,

$$\mathsf{Pr}_{\mathsf{Hyb}}\left[\sum_{s=1}^{N} b_s < k \,\middle|\, \neg E\right] \leq \mathsf{Pr}\left[\sum_{s=1}^{N} Y_s < k\right] \ .$$

Since $\mathsf{Pr}\left[A\right] \leq \mathsf{Pr}\left[A|B\right] \mathsf{Pr}\left[B\right]$, we have that

$$\mathsf{Pr}_{\mathsf{Hyb}}\left[\sum_{s=1}^{N} b_s < k\right] \leq \frac{\mathsf{Pr}\left[\sum_{s=1}^{N} Y_s < k\right]}{\mathsf{Pr}_{\mathsf{Hyb}}\left[E\right]} \leq \mathsf{Pr}\left[\sum_{s=1}^{N} Y_s < k\right] + 2N/T + O(N\sqrt{\delta} + (N/T)^2) \ .$$

The lemma then follows by trace distance. □

5.2 Description of the extractor

For a measurement channel $\mathsf{M} \colon \mathbf{S}(\mathcal{A}) \to \mathbf{S}(\mathcal{A} \otimes \mathcal{O})$, we denote by $\overline{\mathsf{M}} \in U(\mathcal{A} \otimes \mathcal{O} \otimes \mathcal{B})$ some unitary dilation of M. We denote by $\underline{\mathsf{M}} \colon \mathbf{S}(\mathcal{A} \otimes \mathcal{B}) \to \mathbf{S}(\mathcal{A} \otimes \mathcal{O} \otimes \mathcal{B})$ a *projective* dilation of M, given by

$$\underline{\mathsf{M}}(\rho) \coloneqq \sum_i \overline{\mathsf{M}}^{\dagger} \left|i\middle\rangle\!\middle\langle i\right|_{\mathcal{O}} \overline{\mathsf{M}} \rho \overline{\mathsf{M}}^{\dagger} \left|i\middle\rangle\!\middle\langle i\right|_{\mathcal{O}} \overline{\mathsf{M}}$$

where $\{|i\rangle\}_i$ is a basis for $\mathcal{O}$. All of our procedures and correctness analyses are independent of the choice of dilation, and we assume that the circuit complexity of $\overline{\mathsf{M}}, \underline{\mathsf{M}}$ is linear in the circuit complexity of M.

We now describe the extractor, which is a measurement channel $\mathsf{Extract}_{i,\nu} \colon \mathbf{S}(\mathcal{A} \otimes \mathcal{Z}) \to \mathbf{S}(\mathcal{A} \otimes \mathcal{Z} \otimes \mathcal{O})$, where $\mathcal{Z} = (\mathcal{Z}_1, \dots, \mathcal{Z}_t, \mathcal{W}_{t+1})$ are the prover's output registers. Recall that we model the prover as a sequence of unitaries $U_1, \dots, U_t$.

For $i \in [t]$, denote by $U^{(i)} \colon R_i \times \cdots \times R_t \to U(\mathcal{A} \otimes \mathcal{Z}_{i+1} \otimes \cdots \otimes \mathcal{Z}_t \otimes \mathcal{W}_{t+1})$ the map

$$U^{(i)}(r_i, \dots, r_t) = U_t(r_t) \cdots U_i(r_i) \ .$$

For $i \in [t], \mathbf{r} = (r_1, \dots, r_{i-1}) \in R_1 \times \cdots \times R_{i-1}$, let $f^{(i)}_{(\mathbf{r})} \colon (R_i \times \cdots \times R_t) \times (Z_1 \times \cdots \times Z_t \times W_{t+1}) \to \{0,1\}$ denote the function $f^{(i)}_{(\mathbf{r})}(r_i, \dots, r_t, z_1, \dots, z_t, w_{t+1}) \coloneqq V(z_1, r_1, \dots, z_t, r_t, w_{t+1})$.

```
$\mathsf{Extract}_{i,\nu}(r_1, \ldots, r_{i-1})$:

  1. Set $N := \lceil 2t \ln(1/\delta)/\nu^2 \rceil$, $\varepsilon := \nu/4kNt$.
  2. Compute $p_0 \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)}, f_{\mathbf{r}}^{(i)}}$. If $p_0 < \gamma := \sum_{j=i}^{t} \frac{k-1}{R_i} + \nu$, stop and output $\bot$.
  3. For $j = 1, \ldots, k$:
     (a) Set $b := 0$.
     (b) For $s = 1, \ldots, N$, apply the following steps:
           i. Compute $p_s \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)}, f_{\mathbf{r}}^{(i)}}$.
          ii. Choose $r_i \leftarrow R_i \setminus \mathrm{Supp}(W)$ uniformly at random and apply $U_i(r_i)$.
         iii. Initalize ancilla register $\mathcal{B}$ (for $\underline{\mathsf{Threshold}}$) to $|0\rangle$.
          iv. Measure $b \leftarrow \underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)}, f^{(i+1)}}$, where $\gamma' := \sum_{j=i+1}^{t} \frac{k-1}{R_i} + \nu \cdot \frac{t-i-1}{t-i} + \varepsilon$. If
              $b = 1$, go to Step 3c.
           v. Apply $U_i(r_i)^\dagger$.
          vi. Run $\mathsf{Repair}_{kN/2\beta^2}[\mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)}, f_{\mathbf{r}}^{(i)}}, (U_i(r_i))^\dagger \cdot \underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)}, f^{(i+1)}} \cdot U_i(r_i)]$.
     (c) Apply $\overline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)}, f^{(i+1)}}$.
     (d) Compute $\mathcal{W}_{i+1} \leftarrow \overline{\mathsf{Extract}}_{i+1,\nu'}(r_1, \ldots, r_i)$ coherently, for $\nu' := \nu \cdot \frac{t-i-1}{t-i}$.
     (e) If $b = 1$, measure $b' \leftarrow V_i(\mathcal{Z}_1, r_1, \ldots, \mathcal{Z}_i, r_i, \mathcal{W}_i)$.
     (f) If $b = b' = 1$, measure $w_{i+1} \leftarrow \mathcal{W}_{i+1}$ and add $(r_i \mapsto w_i)$ to $W$.
     (g) Apply $\overline{\mathsf{Extract}}_{i+1,\nu'}(r_1, \ldots, r_i)^\dagger$.
     (h) Apply $(\overline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)}, f^{(i+1)}})^\dagger$, then $U_i(r_i)^\dagger$.
     (i) Run $\mathsf{Repair}_{kN/2\beta^2}[\mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)}, f_{\mathbf{r}}^{(i)}}, (U_i(r_i))^\dagger \cdot \underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)}, f^{(i+1)}} \cdot U_i(r_i)]$.
  4. Output $w_i \leftarrow E_i(W)$.
```

$\mathsf{Extract}_{t,\nu}(r_1, \ldots, r_{t-1})$ is simply the [CMSZ21] extractor, modified to sample r_t without replacement:

```
$\mathsf{Extract}_{t,\nu}(r_1, \ldots, r_{t-1})$:

  1. Compute $p_0 \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)}, f_{\mathbf{r}}^{(i)}}$. If $p_0 < \gamma := \frac{k-1}{R_t} + \nu$, stop and output 0.
  2. For $j = 1, \ldots, k$:
     (a) Set $b := 0$.
     (b) For $s = 1, \ldots, N$, and while $b = 0$, apply each of the following steps:
           i. Compute $p_s \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U_t, f_{\mathbf{r}}^{(t)}}$.
          ii. Choose $r_t \leftarrow R_t \setminus \mathrm{Supp}(W)$ uniformly at random and apply $U_t(r_t)$.
         iii. Measure $b \leftarrow V_t(\mathcal{Z}_1, r_1, \ldots, \mathcal{Z}_t, r_t, \mathcal{W}_{t+1})$.
          iv. If $b = 1$, measure $w_{t+1} \leftarrow \mathcal{W}_{t+1}$ and add $(r_t \mapsto w_{t+1})$ to $W$.
           v. Apply $U_t(r_t)^\dagger$.
          vi. Run $\mathsf{Repair}_{kN/2\beta^2}[\mathsf{Estimate}_{\varepsilon,\delta}^{U_t, f_{\mathbf{r}}^{(t)}}, (U_t(r_t))^\dagger \cdot \mathit{\Pi}_{V_t(\mathbf{r},\cdot)} \cdot U_i(r_i)]$.
  3. Output $E_t(W)$.
```

Lemma 4. $\mathsf{Extract}_{i,\nu}$ *is a circuit of size* $P(t, k, \log(1/\delta), 1/\nu) \cdot (ck)^{t-i}$ *for some polynomial* P *and constant* c. *In particular, if* $k = O(1)$, $t = O(\log n)$, $\delta = 2^{-\lambda}$ *and* $\nu = 1/\mathsf{poly}(\lambda)$ *then* $\mathsf{Extract}_\nu = \mathsf{Extract}_{1,\nu}$ *is a polynomial-size quantum circuit.*

Proof. Let P be a polynomial (with positive coefficients) such that for any i,

$$|\mathsf{Extract}_{i,\nu}| \leq P(t, k, \log(1/\delta), 1/\beta, 1/\nu) + k \cdot 2|\mathsf{Extract}_{i+1,\nu'}| \; .$$

Such a polynomial exists by Lemma 2 and Theorem 3. Let c be a constant such that $P(1,1,1,1,2) \leq c \cdot P(1,1,1,1,1)$. The circuit size of $\mathsf{Extract}_{i,\nu}$ is then bounded by

$$\begin{aligned} & P(t,k,\log(1/\delta),1/\beta,1/\nu) + k \cdot 2|\mathsf{Extract}_{i+1,\nu'}| \\ & \leq P(t,k,\log(1/\delta),1/\beta,1/\nu) + 2c^{t-i-1}k^{t-i} \cdot P(t,k,\log(1/\delta),1/\beta,\tfrac{1}{\nu} \cdot \tfrac{t-i}{t-i-1}) \\ & \leq (ck)^{t-i} P(t,k,\log(1/\delta),1/\beta,1/\nu) \ , \end{aligned}$$

since $\frac{t-i}{t-i-1} \leq 2$ for all $i \in \{1,\ldots,t-1\}$. □

5.3 Correctness

The key lemma which establishes the correctness of the extractor is the following.

Lemma 5. *Let* $\mathsf{Extract}'$ *be as* $\mathsf{Extract}$*, except that its output is* 0 *if* $\mathsf{Extract}$ *outputs* $\perp$ *and*

$$V_{i-1}(z_1, r_1, \ldots, z_{i-1}, r_{i-1}, w_{i-1})$$

otherwise. Then for $\gamma := \sum_{j=i}^{t} \frac{k-1}{R_i} + \nu$ *and all* $\mathbf{r} = (r_1, \ldots, r_{i-1})$*,*

$$d_{\mathsf{comp}}^{\mathcal{A},\mathcal{Z},\mathcal{O}}(\underline{\mathsf{Extract}}'_{i,\nu}(\mathbf{r};\rho), \underline{\mathsf{Threshold}}_{\gamma,\varepsilon,\delta}^{U^{(i)},f_{\mathbf{r}}^{(i)}}(\rho)) \leq k^{t-i} \cdot (\beta + O(\beta^2) + \mathsf{poly}(\lambda) \cdot \sqrt[4]{\delta}) + \mathsf{negl}(\lambda) \ .$$

Before proving the lemma, we show how to use it to prove Theorem 2.

Proof (Theorem 2). Let $P^* = (U_1, \ldots, U_t, \rho)$ be an adversary for Π_t. By Lemma 2, $\mathbb{E}[\mathsf{Estimate}_{\varepsilon,\delta}^{U^{(1)},f^{(1)}}(\rho)] = \mathsf{Pr}\left[\langle P^*, V_t \rangle \to 1\right]$. Hence by Proposition 1, $\mathsf{Pr}\left[\mathsf{Threshold}_{\gamma,\varepsilon,\delta}^{U^{(1)},f^{(1)}}(\rho) \to 1\right] \geq \mathsf{Pr}\left[\langle P^*, V_t \rangle \to 1\right] - \gamma$. It follows by Lemma 5 that

$$\mathsf{Pr}\left[\mathsf{Extract}'_{1,\nu/2} \to 1\right] \geq \mathsf{Pr}\left[\langle P^*, V_t \rangle \to 1\right] - \kappa - \nu$$

for $\kappa := \sum_{i=1}^{t} \frac{k-1}{|R_i|}$ and choosing $\beta = \nu/2k^t$. The theorem follows by noting that, by definition, the probability that $\mathsf{Extract}$ succeeds is equal to the probability that $\mathsf{Extract}'$ outputs 1. □

Proof. We argue the inductive step. The base case follows by a similar (simpler) argument.

Consider a hybrid extractor Hyb_1 in which we replace Steps 3f and 4 with

3f'. If $b' = 1$, add $(r_i \mapsto \perp)$ to W.
4'. Output 1 if $|W| = k$, else 0.

By last-round collapsing, $d_{\mathsf{comp}}^{\mathcal{A},\mathcal{Z},\mathcal{O}}(\mathsf{Extract}'_{i,\gamma,\varepsilon}, \mathsf{Hyb}_1) = \mathsf{negl}(\lambda)$.

Observe that after removing the measurement of $\mathcal{W}_i$ in $\mathsf{Extract}'$, Steps 3d, 3e and 3g are equivalent to an invocation of $\underline{\mathsf{Extract}}'_{i+1,\nu'}$. We can now invoke the inductive hypothesis. Specifically, we consider another hybrid extractor Hyb_2, in which we replace Steps 3d to 3g with the following:

– If $b = 1$, measure $b' \leftarrow \underline{\mathsf{Threshold}}_{\gamma'-\varepsilon,\varepsilon,\delta}^{U_{\mathbf{r}},f_{\mathbf{r}}}$. If $b' = 1$, add $(r_i \to \perp)$ to W.

By induction and the triangle inequality, $d(\mathsf{Hyb}_1, \mathsf{Hyb}_2) \leq k^{t-i} \cdot (\varepsilon + O(\varepsilon^2) + \mathsf{poly}(\lambda) \cdot \sqrt[4]{\delta} + \mathsf{negl}(\lambda))$.

Hyb_3 is obtained from Hyb_2 by replacing Step 5.3 with

5.3' If $b = 1$, add $(r_i \to \perp)$ to W.

If $b = 1$, then by Lemma 2, $\mathsf{Pr}\left[b' = 1\right] \geq 1 - \delta$. Hence by gentle measurement, $d^{\mathcal{A},\mathcal{Z},\mathcal{O}}(\mathsf{Hyb}_2, \mathsf{Hyb}_3) = O(k\sqrt{\delta})$. We write out Hyb_3 in full, simplifying where possible.

```
$\mathsf{Hyb}_3$:

1. Compute $p_0 \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)},f_{\mathbf{r}}^{(i)}}$. If $p_0 < \gamma$, stop and output 0.
2. For $j = 1, \ldots, k$:
   (a) Set $b := 0$.
   (b) For $s = 1, \ldots, N$, and while $b = 0$, apply each of the following steps:
       i. Compute $p_s \leftarrow \mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)},f_{\mathbf{r}}^{(i)}}$.
       ii. Choose $r_i \leftarrow R_i \setminus \mathrm{Supp}(W)$ uniformly at random and apply $U_i(r_i)$.
       iii. Initalize ancilla register $\mathcal{B}$ to $|0\rangle$.
       iv. Measure $b \leftarrow \underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)},f^{(i+1)}}$, where $\gamma' := \sum_{j=i+1}^{t} \frac{k-1}{R_i} + \nu \cdot \frac{t-i-1}{t-i} + \varepsilon$.
       v. If $b = 1$, add $(r_i \mapsto \bot)$ to $W$.
       vi. Apply $U_i(r_i)^\dagger$.
       vii. Run $\mathsf{Repair}_{kN/2\beta^2}[\mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i)},f_{\mathbf{r}}^{(i)}}, (U_i(r_i))^\dagger \cdot \underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)},f^{(i+1)}} \cdot U_i(r_i)]$.
3. Output 1 if $|W| = k$, else 0.
```

Consider now the j-th iteration of the outer loop. We compute the quantity ζ from Lemma 3. Let $|v\rangle \in \mathrm{im}(\mathit{\Pi}_{\geq\alpha}^{U^{(i)},f_{\mathbf{r}}^{(i)}})$. Then

$$\mathbb{E}_{r_i \leftarrow R_i \setminus \mathrm{Supp}(W)}[\mathsf{Estimate}_{\varepsilon,\delta}^{U^{(i+1)},f^{(i+1)}}(U_i(r_i)\,|v\rangle)] \geq \langle v|\, E_{U^{(i)},f_{\mathbf{r}}^{(i)}}\, |v\rangle - \frac{j-1}{|R_i|} \geq \alpha - \frac{k-1}{|R_i|}\ .$$

So by Proposition 1,

$$\mathsf{Pr}_{r_i \leftarrow R_i \setminus \mathrm{Supp}(W)}\left[\underline{\mathsf{Threshold}}_{\gamma',\varepsilon,\delta}^{U^{(i+1)},f^{(i+1)}}(U_i(r_i)\,|v\rangle) \rightarrow 1\right] \geq \alpha - \frac{k-1}{|R_i|} - \gamma'.$$

Since we abort if $p_0 < \gamma$, by our choice of ε we have that $\alpha - \frac{k-1}{|R_i|} \geq \gamma' + \frac{\nu}{2t}$. Hence $\zeta \geq \nu/2t$.

Then by Lemma 3, the probability that b is never set to 1 is at most

$$(1 - \nu/2t)^N + O(N(1/T + \sqrt{\delta})) \leq \beta^2/2k + O(N\sqrt{\delta} + \beta^4/k^2)$$

given our choice of N. Hence the probability that $p_0 \geq \gamma$ and Hyb_3 outputs 0 is at most $\beta^2/2 + O(kN\sqrt{\delta} + \beta^4)$. By gentle measurement, $d^{\mathcal{A},\mathcal{Z},\mathcal{B}}(\underline{\mathsf{Hyb}}_3, \underline{\mathsf{Threshold}}_{\gamma,\varepsilon,\delta}^{U^{(i)},f_{\mathbf{r}}^{(i)}}) \leq \beta + O(\sqrt{kN}\sqrt[4]{\delta} + \beta^2)$. The lemma then follows by the triangle inequality. □

6 Collapsing Hash Function Families

In the following, we show that the hash functions $\mathsf{Hash}_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} \bmod q$, indexed by the matrix $\mathbf{A}$, are collapsing and even when $\mathbf{A}$ is "folded" with coefficients being small units in the base ring.

6.1 Definitions

We recall the definition of a hash function family and the desired properties.

Definition 7 (Hash Function Family). *Let $\ell, k \in \mathsf{poly}(\lambda)$. A hash function family $\mathsf{Hash} = (\mathsf{Setup}, \mathsf{H})$ from $\mathcal{X}^\ell$ to $\mathcal{Y}^h$ consists of a PPT Setup algorithm and a deterministic polynomial-time H algorithm. The Setup algorithm inputs a security parameter 1^λ and outputs the public parameters pp. The H algorithm inputs pp and a preimage $\mathbf{x} \in \mathcal{X}^\ell$. It outputs an image $\mathbf{y} \in \mathcal{Y}^h$. When it is clear from the context, we omit the input pp and write $\mathbf{y} = \mathsf{H}(\mathbf{x})$.*

We define below the notion of collapsing for hash functions [Unr16].

Definition 8 (Collapsing). *Let $\ell, k \in \mathsf{poly}(\lambda)$ and $\mathcal{W} \subseteq \mathcal{X}$. Let $\mathsf{Hash} = (\mathsf{Setup}, \mathsf{H})$ be a hash function from $\mathcal{X}^\ell$ to $\mathcal{Y}^h$. We say that Hash is collapsing over $\mathcal{W}^\ell$ if for any efficient (quantum) adversary A*

$$\left|\Pr\left[\mathsf{Collapsing}^0_A(1^\lambda) = 1\right] - \Pr\left[\mathsf{Collapsing}^1_A(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{Collapsing}^b_A$ is defined as follows:

$\underline{\mathsf{Collapsing}^b_A(1^\lambda)}$*:*

1. *Sample pp using the $\mathsf{Setup}(1^\lambda)$ algorithm and send it over to A.*
2. *A replies with a classical bitstring y and a quantum state on a register $\mathcal{X}$.*
3. *Let $U_{\mathsf{H},y}$ be the unitary that acts on $\mathcal{X}$ and a fresh ancilla, and CNOTs into the fresh ancilla the bit that determines whether the output of $\mathsf{H}(\cdot)$ equals y and the input belongs to $\mathcal{W}^\ell$. Apply $U_{\mathsf{pp},y}$, measure the ancilla, and apply $U^\dagger_{\mathsf{pp},y}$.*
4. *If the output of the measurement is 0, then abort the experiment. Else proceed.*
5. *If $b = 0$ do nothing.*
6. *If $b = 1$ measure the register $\mathcal{X}$ in the computational basis, discard the result.*
7. *Return to A all registers and output whichever bit A outputs.*

Note that the security experiment $\mathsf{Collapsing}^b_A$ in the definition of collapsing is a quantum algorithm. It is often easier to work with the classical security notion of somewhere-statistically binding (SSB), defined below, which is known to imply collapsing.

Definition 9 (Somewhere-Statistically Binding). *Let $h, \ell \in \mathsf{poly}(\lambda)$ and $\mathcal{W} \subseteq \mathcal{X}$. A hash function family $\mathsf{Hash} = (\mathsf{Setup}, \mathsf{H})$ from $\mathcal{X}^\ell$ to $\mathcal{Y}^h$ is said to be somewhere-statistically binding (SSB) over $\mathcal{W}^\ell$ if there exists a PPT BSetup algorithm such that the following hold:*

- *The BSetup algorithm inputs a security parameter 1^λ and an index $i \in \mathbb{Z}_\ell$. It outputs the public parameters pp.*
- *For all $i \in \mathbb{Z}_\ell$, the distributions $\mathsf{Setup}(1^\lambda)$ and $\mathsf{BSetup}(1^\lambda, i)$ are computationally indistinguishable.*
- *For all $i \in \mathbb{Z}_\ell$,*

$$\Pr\left[\exists\ \mathbf{x}_0, \mathbf{x}_1 \in \mathcal{W}^\ell : x_{0,i} \neq x_{1,i} \wedge \mathsf{H}(\mathsf{pp}, \mathbf{x}_0) = \mathsf{H}(\mathsf{pp}, \mathbf{x}_1) \ \middle|\ \mathsf{pp} \leftarrow \mathsf{BSetup}(1^\lambda, i)\right] \leq \mathsf{negl}(\lambda).$$

Lemma 6 ([Ma20,ACL+22]). *Let $\mathsf{Hash} = (\mathsf{Setup}, \mathsf{H})$ be a hash function family from $\mathcal{X}^\ell$ to $\mathcal{Y}^h$ and $\mathcal{W} \subseteq \mathcal{X}$. If Hash is SSB over $\mathcal{W}^\ell$, then Hash is collapsing over $\mathcal{W}^\ell$.*

6.2 Bounded Homomorphic Public-Key Encryption

We recall the notion of public-key encryption. Note that we define a variant of public-key encryption with perfect correctness.

Definition 10 (Public-Key Encryption). *A public-key encryption $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ consists of a key generation algorithm Gen that takes as input the security parameter 1^λ and returns a key pair $(\mathsf{pk}, \mathsf{sk})$. The encryption algorithm Enc takes as input pk and a message m an produces a ciphertext c. We require that for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{sk})$ in the support of $\mathsf{Gen}(1^\lambda)$ and all messages m, it holds that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$.*

To prove the security of the hash function family $f_{\mathbf{A}}$ we assume the existence of a bounded linearly homomorphic encryption scheme, that we define in the following.

Definition 11 ((ℓ, β)-Bounded Linearly Homomorphic Encryption over $\mathcal{R}^h_q$). *Let $h, q \in \mathbb{N}$. An encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (ℓ, β)-bounded linearly homomorphic over $\mathcal{R}^h_q$ if the following hold:*

- *(Ciphertext Indistinguishability) For a uniformly sampled key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$, and for all bits $b \in \{0, 1\}$ it holds that the following distributions are computationally indistinguishable:*

$$\mathbf{c} \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, b) \approx \mathbf{u} \leftarrow\!\$\ \mathcal{R}^h_q.$$

– *(Bounded Homomorphism) For all key pairs* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{Gen}(1^\lambda)$*, all bits* $(b_1, \ldots, b_\ell) \in \{0,1\}^\ell$*, all ciphertexts* $(\mathbf{c}_1, \ldots, \mathbf{c}_\ell) \in \mathcal{R}_q^{h \times \ell}$ *in the support of* $(\mathsf{Enc}(\mathsf{pk}, b_1), \ldots, \mathsf{Enc}(\mathsf{pk}, b_\ell))$*, and all vectors* $\mathbf{x} \in \mathcal{R}^\ell$ *where* $\|\mathbf{x}\| \le \beta$*, it holds that:*

$$\mathsf{Dec}(\mathsf{sk}, (\mathbf{c}_1, \ldots, \mathbf{c}_\ell) \cdot \mathbf{x} \bmod q) = \sum_{i=1}^{\ell} b_i \cdot \mathbf{x}_i.$$

Examples of encryption schemes that satisfy the above property are NTRU [HPS98,SS11] (for $h = 1$) and Regev encryption based on (Ring)-LWE [Reg05,LPR10] (for $h > 1$).

6.3 A Fold-Collapsing Hash Function

Let $h, t \in \mathbb{N}$, $\ell = 2^t$, $i \in \{0, 1, \ldots, t\}$, and $(r_j)_{j \in [i]} \in \mathcal{R}^i$. Define $\ell_i := \ell/2^i = 2^{t-i}$. For any matrix $\mathbf{A}_i \in \mathcal{R}_q^{h \times \ell_i}$, we denote by $(\mathbf{A}_{i,0}, \mathbf{A}_{i,1}) \in (\mathcal{R}_q^{h \times \ell_{i+1}})^2$ an arbitrary fixed partitioning of the columns of $\mathbf{A}$ into two disjoint sets of columns of identical cardinality. Similarly, for any vector $\mathbf{x}_i \in \mathcal{R}^{\ell_i}$, we denote by $(\mathbf{x}_{i,0}, \mathbf{x}_{i,1}) \in (\mathcal{R}_q^{\ell_{i+1}})^2$ the partitioning of $\mathbf{x}$ induced by that of $\mathbf{A}$. In Figure 2 we define a hash function family $\mathsf{Hash}_i := \mathsf{Hash}[h, \ell, (r_j)_{j \in [i]}]$ from $\mathcal{X}^{\ell_i}$ to $\mathcal{Y}^h$.

```
$\mathsf{Hash}_i.\mathsf{Setup}(1^\lambda)$
if $i = 0$ then return $\mathbf{A}_0 \leftarrow\$ \mathcal{R}_q^{h \times \ell}$
else $\mathbf{A}_{i-1} \leftarrow \mathsf{Hash}_{i-1}.\mathsf{Setup}(1^\lambda)$
$\mathbf{A}_i := r_i \cdot \mathbf{A}_{i-1,0} + \mathbf{A}_{i-1,1} \bmod q$
return $\mathsf{pp} := \mathbf{A}_i$
```

```
$\mathsf{Hash}_i.\mathsf{H}(\mathbf{x}_i)$
return $\mathbf{y} := \mathbf{A}_i \cdot \mathbf{x}_i \bmod q$
```

Fig. 2. Construction of hash function families Hash_i from $\mathcal{X}^{\ell_i}$ to $\mathcal{Y}^h$, where $\ell_i := \ell/2^i = 2^{t-i}$. For $i = 0$, we denote the family by $\mathsf{Hash}_0 = \mathsf{Hash}[h, \ell]$.

We are now ready to show that the hash function as defined above is SSB, generalizing a Theorem from [ACL+22]. As an immediate corollary, we obtain that the hash function is also collapsing.

Lemma 7 (Collapsing). *Let* $\beta_0 \in \mathbb{R}$*. Let* $\mathcal{W}_0 := \{x \in \mathcal{R} : \|x\| \le \beta_0\}$*. If there exists an* (ℓ, β_0)*-bounded linearly homomorphic encryption over* $\mathcal{R}_q^h$*, then* $\mathsf{Hash}[h, \ell]$ *is SSB over* $\mathcal{W}_0$*.*

Proof. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an (ℓ, β_0)-bounded linearly homomorphic encryption over $\mathcal{R}_q^h$. Let $\mathbf{A}$ be a uniformly sampled hash key. We define ℓ hybrid distributions where we gradually substitute the columns of $\mathbf{A}$ with encryptions of 0. That is, in the i-th hybrid, the key of the hash function consists of

$$(\mathbf{c}_1, \ldots, \mathbf{c}_i, \mathbf{B}_i)$$

where $(\mathbf{c}_i \ldots \mathbf{c}_i) \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, 0)$ and $\mathbf{B}_i \leftarrow\$ \mathcal{R}_q^{h \times (\ell - i)}$. It is easy to show that the hybrids of each neighbouring pair are computationally indistinguishable by the ciphertext indistinguishability of the encryption scheme. Note that, in the ℓ-th hybrid, the hash key consists of a concatenation of encryption of 0.

We will now show that the hash function defined in the ℓ-th hybrid is SSB, and the lemma statement will follow. We define $\mathsf{BSetup}(1^\lambda, i)$ to be identical to the distribution above, except that we substitute the i-th column of the key with $\mathbf{c}_i \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, 1)$. The two distributions are computationally indistinguishable by another application of ciphertext indistinguishability. We now show that there does not exist a pair $(\mathbf{x}_0, \mathbf{x}_1) \in \mathcal{W}_0^{2\ell}$ such that $\mathsf{H}(\mathbf{x}_0) = \mathsf{H}(\mathbf{x}_1)$ and $x_{0,i} \neq x_{1,i}$. Assume towards contradiction that it exists, then we have that

$$\tilde{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_\ell) \cdot \mathbf{x}_0 = (\mathbf{c}_1, \ldots, \mathbf{c}_\ell) \cdot \mathbf{x}_1 \bmod q.$$

By the (ℓ, β_0)-bounded linear homomorphism of the encryption scheme, it holds that $\tilde{c}$ decrypts to two different $x_{0,i}$ and $x_{1,i}$. This contradicts the correctness of the scheme. □

Next we show that the function remains collapsing even if we fold the hashing key by linear combinations with short units. We refer to this property as *fold-collapsing.*

Lemma 8 (Fold-Collapsing). *Let $\beta_i \in \mathbb{R}$, $i \in \mathbb{Z}_t$, $r_{i+1} \in \mathcal{R}^\times$ be a unit with $\|r_{i+1}\| = 1$, $\mathcal{W}_i := \{x \in \mathcal{R} : \|x\| \leq \beta_i\}$, $\mathcal{W}_{i+1} := \left\{x \in \mathcal{R} : \|x\| \leq \gamma_\mathcal{R}^{-1} \cdot \beta_i\right\}$, $\mathsf{Hash}_i := \mathsf{Hash}[h, \ell, (r_j)_{j \in [i]}] = (\mathsf{Setup}_i, \mathsf{H}_i)$, and $\mathsf{Hash}_{i+1} := \mathsf{Hash}[h, \ell, (r_j)_{j \in [i+1]}] = (\mathsf{Setup}_{i+1}, \mathsf{H}_{i+1})$. If Hash_i is SSB over $\mathcal{W}_i^{\ell_i}$, then Hash_{i+1} is SSB over $\mathcal{W}_{i+1}^{\ell_{i+1}}$.*

Proof. Since Hash_i is SSB over $\mathcal{W}_i^{\ell_i}$, there exists a PPT algorithm BSetup_i such that

1. BSetup_i inputs 1^λ and $j \in \mathbb{Z}_{\ell_i}$ and outputs pp.
2. For any $j \in \mathbb{Z}_{\ell_i}$, $\mathsf{Setup}_i(1^\lambda)$ and $\mathsf{BSetup}_i(1^\lambda, j)$ are computationally indistinguishable.
3. For any $j \in \mathbb{Z}_{\ell_i}$,

$$\Pr\left[\exists \mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \mathcal{W}_i^{\ell_i} : x_{i,0,j} \neq x_{i,1,j} \wedge \mathbf{A}_i \cdot \mathbf{x}_{i,0} = \mathbf{A}_i \cdot \mathbf{x}_{i,1} \bmod q \;\middle|\; \mathbf{A}_i \leftarrow \mathsf{BSetup}_i(1^\lambda, j)\right] \leq \mathsf{negl}(\lambda)$$

We construct a PPT algorithm BSetup_{i+1} which, on input $j' \in \mathbb{Z}_{\ell_{i+1}}$, samples $b \in \{0,1\}$, runs BSetup_i on $j = j' + b \cdot \ell_{i+1}$ to obtain $\mathbf{A}_i$, and returns $\mathbf{A}_{i+1} \coloneqq r_{i+1} \cdot \mathbf{A}_{i,0} + \mathbf{A}_{i,1} \bmod q$. By Property 2 above, we clearly have that $\mathsf{Setup}_{i+1}(1^\lambda)$ and $\mathsf{BSetup}_{i+1}(1^\lambda, j')$ are computationally indistinguishable for all $j' \in \mathbb{Z}_{\ell_{i+1}}$.

Fix any $j \in \mathbb{Z}_{\ell_i}$ and $j' \in \mathbb{Z}_{\ell_{i+1}}$ satisfying $j = j' \bmod \ell_{i+1}$, any $\mathbf{A}_i \in \mathsf{BSetup}_i(1^\lambda, j)$, any $\mathbf{A}_{i+1} = r_{i+1} \cdot \mathbf{A}_{i,0} + \mathbf{A}_{i,1} \bmod q \in \mathsf{BSetup}_{i+1}(1^\lambda, j')$, and any $\mathbf{x}_{i+1,0}, \mathbf{x}_{i+1,1} \in \mathcal{W}_{i+1}^{\ell_{i+1}}$ satisfying $\mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,0} = \mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,1} \bmod q$. Define $\mathbf{x}_{i,0} := (r_{i+1} \cdot \mathbf{x}_{i+1,0}, \mathbf{x}_{i+1,0})$ and $\mathbf{x}_{i,1} = (r_{i+1} \cdot \mathbf{x}_{i+1,1}, \mathbf{x}_{i+1,1})$.

Note that $\|\mathbf{x}_{i+1,0}\| \leq \gamma_\mathcal{R}^{-1} \cdot \beta_i$ and $\|\mathbf{x}_{i+1,1}\| \leq \gamma_\mathcal{R}^{-1} \cdot \beta_i$. Clearly $\|\mathbf{x}_{i,0}\| \leq \beta_i$ and $\|\mathbf{x}_{i,1}\| \leq \beta_i$. In other words, we have $\mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \mathcal{W}_i^{\ell_i}$.

Since $\mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,0} = \mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,1} \bmod q$, we have

$$\begin{aligned}
\mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,0} &= \mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,1} \bmod q, \\
(r_{i+1} \cdot \mathbf{A}_{i,0} + \mathbf{A}_{i,1}) \cdot \mathbf{x}_{i+1,0} &= (r_{i+1} \cdot \mathbf{A}_{i,0} + \mathbf{A}_{i,1}) \cdot \mathbf{x}_{i+1,1} \bmod q, \\
\mathbf{A}_i \cdot (r_{i+1} \cdot \mathbf{x}_{i+1,0}, \mathbf{x}_{i+1,0}) &= \mathbf{A}_i \cdot (r_{i+1} \cdot \mathbf{x}_{i+1,1}, \mathbf{x}_{i+1,1}) \bmod q, \\
\mathbf{A}_i \cdot \mathbf{x}_{i,0} &= \mathbf{A}_i \cdot \mathbf{x}_{i,1} \bmod q.
\end{aligned}$$

Furthermore, if $x_{i+1,0,j'} \neq x_{i+1,1,j'}$, we have $x_{i,0,j'} \neq x_{i,1,j}$ and $x_{i,0,j'+\ell_{i+1}} \neq x_{i,1,j'+\ell_{i+1}}$ since $r_{i+1} \in \mathcal{R}^\times$ is a unit in $\mathcal{R}$.

Suppose Hash_{i+1} is not SSB over $\mathcal{W}_{i+1}^{\ell_{i+1}}$, then there exists $j' \in \mathbb{Z}_{\ell_{i+1}}$ such that

$$\Pr\left[\exists\ \mathbf{x}_{i+1,0}, \mathbf{x}_{i+1,1} \in \mathcal{W}_{i+1}^{\ell_{i+1}} : x_{i+1,0,j'} \neq x_{i+1,1,j'} \wedge \mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,0} = \mathbf{A}_{i+1} \cdot \mathbf{x}_{i+1,1} \bmod q \;\middle|\; \mathbf{A}_{i+1} \leftarrow \mathsf{BSetup}_{i+1}(1^\lambda, j')\right]$$

is non-negligible. Consequently, by the above derivation, the average

$$\begin{aligned}
&\frac{1}{2} \cdot \Pr\left[\exists\ \mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \mathcal{W}_i^{\ell_i} : x_{i,0,j'} \neq x_{i,1,j'} \wedge \mathbf{A}_i \cdot \mathbf{x}_{i,0} = \mathbf{A}_i \cdot \mathbf{x}_{i,1} \bmod q \;\middle|\; \mathbf{A}_i \leftarrow \mathsf{BSetup}_i(1^\lambda, j')\right] \\
+&\frac{1}{2} \cdot \Pr\left[\exists\ \mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \mathcal{W}_i^{\ell_i} : x_{i,0,j'+\ell_{i+1}} \neq x_{i,1,j'+\ell_{i+1}} \wedge \mathbf{A}_i \cdot \mathbf{x}_{i,0} = \mathbf{A}_i \cdot \mathbf{x}_{i,1} \bmod q \;\middle|\; \mathbf{A}_i \leftarrow \mathsf{BSetup}_i(1^\lambda, j' + \ell_{i+1})\right]
\end{aligned}$$

is non-negligible. We conclude that there exists $j \in \{j', j' + \ell_{i+1}\} \subseteq \mathbb{Z}_{\ell_i}$ such that

$$\Pr\left[\exists\ \mathbf{x}_{i,0}, \mathbf{x}_{i,1} \in \mathcal{W}_i^{\ell_i} : x_{i,0,j} \neq x_{i,1,j} \wedge \mathbf{A}_i \cdot \mathbf{x}_{i,0} = \mathbf{A}_i \cdot \mathbf{x}_{i,1} \bmod q \;\middle|\; \mathbf{A}_i \leftarrow \mathsf{BSetup}_i(1^\lambda, j)\right]$$

is non-negligible, contradicting Property 3 above. □

Note that the elements r_j of the sets R defined in Proposition 2 satisfy the requirements in Lemma 8.

7 Bulletproofs

In this section, we recall the family of lattice-based Bulletproofs protocols [BLNS20,AL21,ACK21] and prove that they are recursive special sound and last-round collapsing.

Let $h, \ell, m, q, t = \mathsf{poly}(\lambda)$ with $\ell = 2^t$, $R \subseteq \mathcal{R}$ a finite subset, $s \in \mathcal{R}$ a slack element, and $(\beta_i)_{i=0}^t \in \mathbb{R}^{t+1}$ a sequence of norm bounds. In Fig. 3, we recall the construction of lattice-based Bulletproofs stated as a family of protocols $(\mathsf{Setup}, \Sigma_i)_{i=0}^t$ parametrized by the above parameters. Each Σ_i is a public-coin $(2i+1)$-message protocol associated to the spaces $(\mathcal{R}_q^h, \mathcal{R}^\ell, (\mathcal{R}_q^{2h}, R)^i, \mathcal{R}^{\ell/2^i})$ where both the prover P_i and the verifier V_i input the public parameters consisting of a matrix $\mathbf{A} \in \mathcal{R}_q^{h \times \ell}$ and a statement consisting of a vector $\mathbf{y} \in \mathcal{R}_q^h$. The prover P_i additionally inputs a witness which consists of a vector $\mathbf{x} \in \mathcal{R}^\ell$. Note that Σ_t is the lattice-based Bulletproofs protocol as described in prior works [BLNS20,AL21,ACK21].

Remark 2. It is common in the lattice setting that a soundness gap exists in argument systems, i.e. the relation that the argument is complete for is a subset of the relation that the argument is sound for. For lattice-based Bulletproofs, it means that while the prover is able to convince the verifier about $(\mathbf{A}, \mathbf{y})$ if it has a short preimage $\mathbf{x}$ satisfying $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$, the knowledge extractor would only be able to extract another slightly longer vector $\mathbf{x}'$ satisfying $\mathbf{A} \cdot \mathbf{x}' = s \cdot \mathbf{y} \bmod q$ for some slack element $s \in \mathcal{R}$ depending on the choice of $\mathcal{R}$. Since we mainly focus on the soundness of lattice-based Bulletproofs protocols in this work, we let the verifier in the protocol in Fig. 3 check the relaxed soundness relation instead of the completeness relation.

```
$\Sigma_i = (P_i, V_i)$
----------------------------------------------------------------------------------------------
$P_i(\mathbf{A}, \mathbf{y}, \mathbf{x})$                                   $V_i(\mathbf{A}, \mathbf{y})$
$(\mathbf{A}_0, \mathbf{y}_0) \coloneqq (\mathbf{A}, \mathbf{y})$           $(\mathbf{A}_0, \mathbf{y}_0) \coloneqq (\mathbf{A}, \mathbf{y})$
$\mathbf{x}_0 \coloneqq \mathbf{x}$
for $j \in [i]$ do                                                          for $j \in [i]$ do
  $\mathbf{l}_{j-1} \coloneqq \mathbf{A}_{j-1,1}\ \mathbf{x}_{j-1,0}$         $r_j \leftarrow\$ R$
  $\mathbf{r}_{j-1} \coloneqq \mathbf{A}_{j-1,0}\ \mathbf{x}_{j-1,1}$
                          $\xrightarrow{\mathbf{l}_{j-1}, \mathbf{r}_{j-1}}$
                          $\xleftarrow{r_j}$
  $\mathbf{A}_j \coloneqq r_j \cdot \mathbf{A}_{j-1,0} + \mathbf{A}_{j-1,1}$                    $\mathbf{A}_j \coloneqq r_j \cdot \mathbf{A}_{j-1,0} + \mathbf{A}_{j-1,1}$
  $\mathbf{y}_j \coloneqq \mathbf{l}_{j-1} + r_j \cdot \mathbf{y}_{j-1} + r_i^2 \cdot \mathbf{r}_{j-1}$   $\mathbf{y}_j \coloneqq \mathbf{l}_{j-1} + r_j \cdot \mathbf{y}_{j-1} + r_i^2 \cdot \mathbf{r}_{j-1}$
  $\mathbf{x}_j \coloneqq \mathbf{x}_{j-1,0} + r_j \cdot \mathbf{x}_{j-1,1}$
endfor                                                                      endfor
$\mathbf{x}_i \coloneqq s^{t-i} \cdot \mathbf{x}_i$
                          $\xrightarrow{\mathbf{x}_i}$
                                                                            return $(\mathbf{A}_i \cdot \mathbf{x}_i = s^{t-i} \cdot \mathbf{y}_i \bmod q) \land (\|\mathbf{x}_i\| \leq \beta_i)$
```

Fig. 3. Family of lattice-based Bulletproofs protocols $(\Pi_i)_{i=0}^t = (\mathsf{Setup}, \Sigma_i)_{i=0}^t$ over $\mathcal{R}$ parametrized by a challenge set $R \subseteq \mathcal{R}$, a slack $s \in \mathcal{R}$, and norm bounds $(\beta_i)_{i=0}^t \in \mathbb{R}^{t+1}$, where $\mathsf{Setup}(1^\lambda)$ returns $\mathbf{A}$ sampled uniformly at random from $\mathcal{R}_q^{h \times \ell}$.

It is well-known that the lattice-based Bulletproofs protocol is $(3, 3, \ldots, 3)$-special sound. Analogously, we show that the family of lattice-based Bulletproofs protocols constructed in Fig. 3 is recursive 3-special sound.

Lemma 9. *If m is a power of a prime p, let $R = \left\{1, 1+\zeta, \ldots, \sum_{i \in \mathbb{Z}_{p-1}} \zeta^i\right\}$, $s = 1$, and $(\beta_i)_{i=0}^t$ be such that $\beta_i = 24^{-i} \cdot \varphi(m)^{-i} \cdot \gamma_{\mathcal{R}}^{-i} \cdot \beta_0$ for all $i \in [t]$. If m is a power of 2, let $R = \left\{1, \zeta, \ldots, \zeta^{\varphi(m)-1}\right\}$, $s = 2$, and $(\beta_i)_{i=0}^t$ be such that $\beta_i = 3^{-i} \cdot \varphi(m)^{-i} \cdot \gamma_{\mathcal{R}}^{-i} \cdot \beta_0$ for all $i \in [t]$. In either case, the family of lattice-based Bulletproofs protocols $(\Pi_i)_{i=0}^t$ constructed in Fig. 3 is recursive 3-special sound.*

Proof. For each $i \in [t]$, we construct the following deterministic polynomial-time extractor E_i:

$$E_i((r_i^{(j)}, \mathbf{x}_i^{(j)})_{j \in [3]}) \coloneqq \begin{pmatrix} r_i^{(1)} \cdot \mathbf{x}_i^{(1)} & r_i^{(2)} \cdot \mathbf{x}_i^{(2)} & r_i^{(3)} \cdot \mathbf{x}_i^{(3)} \\ \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \mathbf{x}_i^{(3)} \end{pmatrix} \cdot s \cdot \mathbf{V}_T^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

where $T \coloneqq (r_i^{(1)}, r_i^{(2)}, r_i^{(3)})$. For $r_i^{(j)} \in R$ and $\mathbf{x}_i^{(j)} \in \mathcal{R}^{\ell/2^i}$, Proposition 2 implies that $s \cdot \mathbf{V}_T^{-1} \in \mathcal{R}^{3 \times 3}$, and therefore $E_i((r_i^{(j)}, \mathbf{x}_i^{(j)})_{j \in [3]})$ outputs a vector $\mathbf{x}_{i-1} \in \mathcal{R}^{\ell/2^{i-1}}$.

Fix any $i \in [t]$. Suppose $(\mathbf{A}, \mathbf{y}, (\mathbf{l}_1, \mathbf{r}_1), r_1, \ldots, (\mathbf{l}_{i-1}, \mathbf{r}_{i-1}), r_{i-1}, (\mathbf{l}_i, \mathbf{r}_i), (r_i^{(j)}, \mathbf{x}_i^{(j)})_{j \in [3]})$ is an accepting 3-branch of transcripts for Σ_i. Let $T \coloneqq (r_i^{(1)}, r_i^{(2)}, r_i^{(3)})$ and $\mathbf{x}_{i-1} \coloneqq E_i((r_i^{(j)}, \mathbf{x}_i^{(j)})_{j \in [3]})$. We have

$$\mathbf{A}_i \ \begin{pmatrix} \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \mathbf{x}_i^{(3)} \end{pmatrix} = \mathbf{y}_i \bmod q$$

$$\mathbf{A}_{i-1} \ \begin{pmatrix} r_i^{(1)} \cdot \mathbf{x}_i^{(1)} & r_i^{(2)} \cdot \mathbf{x}_i^{(2)} & r_i^{(3)} \cdot \mathbf{x}_i^{(3)} \\ \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \mathbf{x}_i^{(3)} \end{pmatrix} = \begin{pmatrix} \mathbf{l}_{i-1} & \mathbf{y}_{i-1} & \mathbf{r}_{i-1} \end{pmatrix} \ \mathbf{V}_T \bmod q$$

$$\mathbf{A}_{i-1} \ \begin{pmatrix} r_i^{(1)} \cdot \mathbf{x}_i^{(1)} & r_i^{(2)} \cdot \mathbf{x}_i^{(2)} & r_i^{(3)} \cdot \mathbf{x}_i^{(3)} \\ \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \mathbf{x}_i^{(3)} \end{pmatrix} \cdot s \cdot \mathbf{V}_T^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{l}_{i-1} & \mathbf{y}_{i-1} & \mathbf{r}_{i-1} \end{pmatrix} \ \mathbf{V}_T \cdot s \cdot \mathbf{V}_T^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \bmod q$$

$$\mathbf{A}_{i-1} \ \mathbf{x}_{i-1} = \mathbf{y}_{i-1} \bmod q$$

Furthermore, since $\left\| \mathbf{x}_i^{(j)} \right\| \leq \beta_i$ for all $j \in [3]$, by Proposition 2, we have $\mathbf{x}_{i-1} \leq \beta_{i-1}$. In other words, $(\mathbf{A}, \mathbf{y}, (\mathbf{l}_1, \mathbf{r}_1), r_1, \ldots, r_{i-1}, \mathbf{x}_{i-1})$ is an accepting transcript for Σ_{i-1}. □

Finally, we show that the family of lattice-based Bulletproofs protocols constructed in Fig. 3 is last-round collapsing.

Lemma 10. *Let $\mathcal{R}$, s, and $(\beta_i)_{i=0}^t$ be as in Lemma 9. Furthermore, let $(\hat{\beta}_i)_{i=0}^t$ be such that $\hat{\beta}_i = \gamma_{\mathcal{R}}^{-i} \cdot \beta_0$ for all $i \in [t]$. If there exists an (ℓ, β_0)-bounded linearly homomorphic encryption over $\mathcal{R}_q^\ell$, then for each $i \in \{0, 1, \ldots, t\}$ the lattice-based Bulletproofs protocol Π_i constructed in Fig. 3 is last-round collapsing.*

Proof. The theorem statement holds trivially for $i = 0$. In the following, let $i \in [t]$. Let $\mathcal{W}_{i+1} = \{x \in \mathcal{R} : \|x\| \leq \beta_i\}$ and $\hat{\mathcal{W}}_{i+1} = \left\{x \in \mathcal{R} : \|x\| \leq \hat{\beta}_i\right\}$. We observe that upon receiving the last prover message $\mathbf{x}_i$, the conditions checked by the verifier are $\mathsf{Hash}_i(\mathbf{x}_i) = s^{t-i} \cdot \mathbf{y}_i \bmod q$ and $\mathbf{x}_i \in \mathcal{W}_{i+1}^{\ell_i}$, where $\mathsf{Hash}_i \coloneqq \mathsf{Hash}[r_1, \ldots, r_i]$ with r_j being the j-th challenge sent by the verifier, for some $\mathbf{y}_i$ which is independent of the last-round message $\mathbf{x}_i$. By Lemmas 7 and 8, we have that Hash_i is collapsing over $\hat{\mathcal{W}}_{i+1}$ for all possible choices of $(r_1, \ldots, r_i)$. Since $\mathcal{W}_{i+1} \subseteq \hat{\mathcal{W}}_{i+1}$, Hash_i is also collapsing over $\mathcal{W}_{i+1}$ for all possible choices of $(r_1, \ldots, r_i)$. In other words, the argument system Π_i is last-round collapsing. □

Combining Theorem 2 and Lemmas 9 and 10, we obtain Theorem 4 as the main result of this work.

Theorem 4 (Formal version of Theorem 1). *Let m, $\mathcal{R}$, R, s, and $(\beta_i, \hat{\beta}_i)_{i=0}^t$ be as in Lemmas 9 and 10, where m is either a prime or a power of 2. If there exists an (ℓ, β_0)-bounded linearly homomorphic encryption over $\mathcal{R}_q^\ell$, then the lattice-based Bulletproofs protocol Π_t constructed in Fig. 3 is a post-quantum proof of knowledge for the relation*

$$\left\{(\mathbf{A}, \mathbf{y}, \mathbf{x}) \in \mathcal{R}_q^{h \times \ell} \times \mathcal{R}_q^h \times \mathcal{R}^\ell : \mathbf{A}\mathbf{x} = s^t \cdot \mathbf{y} \bmod q \wedge \|\mathbf{x}\| \leq \beta_0\right\}$$

with knowledge error $2 \log \ell / \varphi(m)$.

Acknowledgments

The authors thank Fermi Ma for many helpful discussions throughout the development of this work.

References

AC20. Thomas Attema and Ronald Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 513–543. Springer, Heidelberg, August 2020.

ACK21. Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 549–579, Virtual Event, August 2021. Springer, Heidelberg.

ACL$^+$22. Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based snarks: Publicly verifiable, preprocessing, and recursively composable. To appear in CRYPTO 2022, 2022.

AF21. Thomas Attema and Serge Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. Cryptology ePrint Archive, Report 2021/1259, 2021. https://ia.cr/2021/1259.

AL21. Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 519–548, Virtual Event, August 2021. Springer, Heidelberg.

BBB$^+$18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

BCC$^+$16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016.

BCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Heidelberg, October / November 2016.

BCS21. Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Sumcheck arguments and their applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 742–773, Virtual Event, August 2021. Springer, Heidelberg.

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

CMSZ21. Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *FOCS*, pages 49–58. IEEE, 2021.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, Heidelberg, June 1998.

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

LMR19. Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. Succinct arguments for bilinear group arithmetic: Practical structure-preserving cryptography. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2057–2074. ACM Press, November 2019.

LMS21. Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited (or: How to do quantum rewinding undetectably). *CoRR*, abs/2111.12257, 2021.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

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

Ma20. Fermi Ma. Quantum-secure commitments and collapsing hash functions. https://www.cs.princeton.edu/~fermim/talks/collapse-binding.pdf, April 2020.

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994.

MW04. Chris Marriott and John Watrous. Quantum arthur-merlin games. In *Computational Complexity Conference*, pages 275–285. IEEE Computer Society, 2004.

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

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

SS11. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.

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

Unr16. Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.

Win99. A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999.

Zha20. Mark Zhandry. Schrödinger's pirate: How to trace a quantum decoder. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 61–91. Springer, Heidelberg, November 2020.

A Composition of Arguments

In this section, we establish a connection between special sound arguments and recursive special sound arguments. Specifically, we observe that a family of special sound arguments with compatible structural properties can be composed to obtain a family of recursive special sound arguments.

Definition 12 (k-Special Soundness). *Let $\Pi = (\mathsf{Setup}, \Sigma = (P, V))$ be a 3-message public-coin argument system associated to the spaces (X, W, Z, R, W'). The system Π is said to be k-special sound for a relation $\mathfrak{R}$ if there exists an efficient extractor E satisfying the following properties:*

- *The extractor E takes as input $(r^{(j)}, w^{(j)})_{j\in[k]} \in (R \times W')^k$ and outputs some value $w \in W$.*
- *If $(\mathsf{pp}, x, z, (r^{(j)}, w^{(j)})_{j\in[k]})$ is an accepting k-branch of transcripts for Σ, and $w = E((r^{(j)}, w^{(j)})_{j\in[k]})$, then $\mathfrak{R}(\mathsf{pp}, x, w) = 1$.*

Definition 13 (Trivial Protocol). *The trivial protocol $\Sigma = (P, V)$ for a relation $\mathfrak{R}$ is a 1-message protocol where $P(\mathsf{pp}, x, w)$ sends w and $V(\mathsf{pp}, x)$ outputs $\mathfrak{R}(\mathsf{pp}, x, w)$.*

The following theorem is immediate by observing the definitions of special soundness and recursive special soundness.

Theorem 5. *Let $\mathfrak{R}$ be a relation, $\Sigma_0 = (P_0, V_0)$ be the trivial argument for $\mathfrak{R}$, and $\Pi_1 = (\mathsf{Setup}, \Sigma_1 = (P_1, V_1))$ be a 3-message public-coin argument system which is k-special sound for $\mathfrak{R}$. Let $\Pi_0 = (\mathsf{Setup}, \Sigma_0)$. The family $(\Pi_i)_{i=0}^{1}$ is recursive k-special sound.*

Definition 14 (Argument System Composition). *Let $(\Pi_i = (\mathsf{Setup}, \Sigma_i = (P_i, V_i)))_{i=0}^{t}$ and $(\Pi'_i = (\mathsf{Setup}', \Sigma'_i = (P'_i, V'_i)))_{i=0}^{t'}$ be families of public-coin argument systems, where Π_i and Π'_i are $(2i+1)$-message, associated to the spaces $(X, W, (Z_j, R_j)_{j\in[i]}, W_{i+1})_{i=0}^{t}$ and $(X', W', (Z'_j, R'_j)_{j\in[i]}, W'_{i+1})_{i=0}^{t'}$ respectively. Suppose Π_t and Π'_0 satisfy the following structural properties:*

- *There exists a polynomial-time* $\mathsf{TransSetup}$ *algorithm such that the ensembles of distributions*

$$\left\{\mathsf{pp}' : \mathsf{pp}' \leftarrow \mathsf{Setup}'(1^\lambda)\right\}_{\lambda\in\mathbb{N}} \qquad \text{and} \qquad \left\{\mathsf{pp}' : \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ r_i \leftarrow R_i,\ \forall i \in [t] \\ \mathsf{pp}' \leftarrow \mathsf{TransSetup}(\mathsf{pp}, r_1, \ldots, r_t) \end{array}\right\}_{\lambda\in\mathbb{N}}$$

  are computationally indistinguishable.

- *$W_t = W'$.*
- *After the $2t$-th message r_t is being sent by $V_t(\mathsf{pp}, x)$, both $P_t(\mathsf{pp}, x, w)$ and $V_t(\mathsf{pp}, x)$ can derive some statement $x_t \in X'$.*
- *After the $(2t+1)$-th message w_t is being sent by $P_t(\mathsf{pp}, x, w)$, $V_t(\mathsf{pp}, x)$ accepts if $V'_0(\mathsf{pp}', x_t)$ accepts w_t, where $\mathsf{pp}' := \mathsf{TransSetup}(\mathsf{pp}, r_1, \ldots, r_t)$ and $r_1, \ldots, r_t$ are the challenges sent by V_t.*

The TransSetup*-compositon of the two families, written as $(\Pi_i)_{i=0}^{t+t'} = (\Pi_i)_{i=0}^{t} \diamond_{\mathsf{TransSetup}} (\Pi'_i)_{i=0}^{t'}$, is a family of public-coin argument systems where, for $j \in [t']$, $\Pi_{t+j} = (\mathsf{Setup}, \Sigma_{t+j} = (P_{t+j}, V_{t+j}))$ is a $(2(t+j)+1)$-message public-coin argument system, where $(P_{t+j}(\mathsf{pp}, x, w), V_{t+j}(\mathsf{pp}, x))$ is constructed as follows:*

- *Execute the interaction $(P_t(\mathsf{pp}, x, w), V_t(\mathsf{pp}, x))$ until the $2t$-th message is sent, with P_{t+j} playing the role of P_t and V_{t+j} playing the role of V_t, so that both P_{t+j} and V_{t+j} derive a statement $x_t \in X'$.*
- *Let $r_1, \ldots, r_t$ be the challenges sampled by V_t, and w_t be the $(2t+1)$-th message supposed to be sent by P_t in Σ_t.*
- *Both P_{t+j} and P_t and V_{t+j} compute $\mathsf{pp}' := \mathsf{TransSetup}(\mathsf{pp}, r_1, \ldots, r_t)$.*
- *Execute the interaction $(P'_j(\mathsf{pp}', x_t, w_t), V'_j(\mathsf{pp}', x_t))$ with P_{t+j} playing the role of P'_j and V_{t+j} playing the role of V'_j.*
- *V_{t+j} outputs whatever V'_j outputs.*

By the construction of the composition in Definition 14, the following theorem is immediate.

Theorem 6. *If $(\Pi_i)_{i=0}^{t}$, $(\Pi'_i)_{i=0}^{t'}$, and* TransSetup *be as in Definition 14, then the family*

$$(\Pi_i)_{i=0}^{t+t'} = (\Pi_i)_{i=0}^{t} \diamond_{\mathsf{TransSetup}} (\Pi'_i)_{i=0}^{t'}$$

is recursive k-special sound. Furthermore, for each $j \in [t]$, if Π'_j is last-round collapsing, then Π_{t+j} is last-round collapsing.

Combining Theorems 5 and 6, we obtain the following corollary.

Corollary 1. *For $i \in [t]$, let $\mathfrak{R}_i$ be a relation, $\Sigma_{i,0}$ be the trivial argument for $\mathfrak{R}_i$, and $\Pi_{i,1} = (\mathsf{Setup}_i, \Sigma_{i,1})$ be a 3-message public-coin argument system which is k-special sound for $\mathfrak{R}_i$. Let $\Pi_{i,0} = (\mathsf{Setup}_i, \Sigma_{i,0})$ for $i \in [t]$. If $(\Pi_{i,j})_{j=0}^{1}$, $(\Pi_{i+1,j})_{j=0}^{1}$, and $\mathsf{TransSetup}_i$ satisfy the conditions in Definition 14 for all $i \in [t-1]$, then the family $(\Pi'_i)_{i=0}^{t} = (\Pi_{1,j})_{j=0}^{1} \diamond_{\mathsf{TransSetup}_1} (\Pi_{2,j})_{j=0}^{1} \diamond_{\mathsf{TransSetup}_2} \cdots \diamond_{\mathsf{TransSetup}_{t-1}} (\Pi_{t,j})_{j=0}^{1}$ is recursive k-special sound. Furthermore, for all $i \in [t]$, if $\Pi_{i,1}$ is last-round collapsing, then Π'_i is last-round collapsing.*

The lattice-based Bulletproofs protocols family described in the fashion of Fig. 3 can be seen as the result of applying Corollary 1 to lattice-based Bulletproofs in the fashion those described in the literature [BLNS20,AL21,ACK21].