Interactive Oracle Arguments in the QROM and Applications to Succinct Verification of Quantum Computation

Islam Faisal[1] ✉
Department of Computer Science, Boston University, USA

Abstract

This work is motivated by the following question: can an untrusted quantum server convince a classical verifier of the answer to an efficient quantum computation using only polylogarithmic communication? We show how to achieve this in the quantum random oracle model (QROM), after a non-succinct instance-independent setup phase.

We introduce and formalize the notion of post-quantum interactive oracle arguments for languages in QMA, a generalization of interactive oracle proofs (Ben-Sasson–Chiesa–Spooner). We then show how to compile any non-adaptive public-coin interactive oracle argument (with private setup) into a succinct argument (with setup) in the QROM.

To conditionally answer our motivating question via this framework under the post-quantum hardness assumption of LWE, we show that the XZ local Hamiltonian problem with at least inverse-polylogarithmic relative promise gap has an interactive oracle argument with instance-independent setup, which we can then compile.

Assuming a variant of the quantum PCP conjecture that we introduce called the *weak XZ quantum PCP conjecture*, we obtain a succinct argument for QMA (and consequently the verification of quantum computation) in the QROM (with non-succinct instance-independent setup) which makes only black-box use of the underlying cryptographic primitives.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols; Theory of computation → Interactive proof systems; Theory of computation → Quantum complexity theory

Keywords and phrases succinct arguments, interactive oracle proofs, delegation of quantum computation, quantum random oracle model, QROM, BQP, QMA

Funding *Islam Faisal*: partially supported by NSF grants CCF-1947889 and CNS-1414119.

Acknowledgements Thanks to Mark Bun and Nicholas Spooner for their mentorship and help with refining and revising this project. Thanks - in alphabetical order - to Adam Smith, Alex Bredariol Grilo, Alex Lombardi, Anand Natarajan, Andrea Coladangelo, Anurag Anshu, Assaf Kfoury, Azer Bestavros, Chris Laumann, Fermi Ma, Ibrahim Faisal, Jiayu Zhang, Leo Reyzin, Ludmila Glinskih, Luowen Qian, Mayank Varia, Muhammad Faisal, Nadya Voronova, Nathan Ju, Ran Canetti, Sam Gunn, Steve Homer, Thomas Vidick, Urmila Mahadev, and others for their time listening to ideas, helpful discussions, and explanations. I specially thank Thomas Vidick for his online lectures that spurred my interest in this research topic as well as his continued support. Thanks to [ACGH20, DFMS22b] for making the LaTeX sources of their papers accessible which helped in typesetting this paper. Thanks to UCLA's Institute for Pure and Applied Mathematics (IPAM) for the support received to participate in the Graduate Summer School on Post-quantum and Quantum Cryptography where I discussed this work with other participants.

1 Introduction

This work is motivated by the following use case which is desirable in a world where quantum computers reach larger scales but are only available in controlled facilities or laboratories.

> **Real World Application:** Alice owns only classical devices (e.g. laptop and/or tablet) and a classical internet connection. She wants to delegate some efficient quantum-computational tasks to a quantum server (Merlin) in a remote location. How can she make sure that the quantum server performed the intended tasks using only a *succinct* amount of classical internet communication?

[1] The author welcomes and encourages feedback about this under-review manuscript to be sent to: islam@bu.edu.

Under some assumptions, we show how this can be achieved after a non-succinct initial setup phase that does not depend on the subsequent tasks to be delegated. In particular, we show the following result.

▶ **Informal Theorem 1** (Informal Statement of Theorem 17). *If a variant of the quantum PCP conjecture (Conjecture 1) is true as well as the post-quantum hardness of LWE, then there exists a classical-verifier succinct-communication argument with non-succinct setup in the QROM for* QMA *(and consequently for the verification of quantum computation).*

The general topic of delegating quantum computation has been studied for a while (for a non-exhaustive list of works, see for example [Chi05, FHM18, GKK18, Mah18b, ACGH20, CCY20, Zha22, TMT22]). In early work, the verifier was modeled as a (possibly weaker) quantum device (e.g. [Chi05]). Mahadev's breakthrough [Mah18a, Mah18b] enabled classical verification of quantum computation under the post-quantum hardness assumption of Learning with Errors (LWE). This opened the door to further subsequent developments in the topic of classical verification of quantum computation (e.g. [VZ19, ACGH20]). In particular, the question of succinct verification of quantum computation has been studied in these works [CCY20, BM22, BKL^{+}22, CM21, GJMZ22, Zha21]. We discuss how they differ from our work in Section 1.1.

We will now go from our motivating question to the more general problem of deciding whether a local Hamiltonian has a low-energy groundstate. The details of the reduction from verification of quantum computation to the local Hamiltonian problem can be found in [FHM18] where the standard Feynman-Kitaev *circuit-to-Hamiltonian* reduction is used. As alluded to in some papers such as [BL08, FHM18], one can obtain XZ Hamiltonians from the Kitaev construction by using a suitable universal gate set [2].

A classical-verifier protocol for the XZ local-Hamiltonian problem has been given in [ACGH20] by iterating on a long sequence of works starting by Kitaev in 1999 and culminating in the recent works of [MNS16, FHM18, MF16, Mah18b, VZ19, CVZ20]. We modify the protocol to eliminate redundant communication. Then we identify the modified protocol as an instance of an *interactive oracle argument*, a concept that we define by generalizing interactive oracle proofs [BCS16].

Post-quantum interactive oracle arguments - which we define in this paper - are interactive protocols for yes/no promise problems where yes instances are defined by a quantum-witness relation. In this class of protocols, prover messages are modeled as *oracles* that can be query-accessed by the verifier. Our main technical contribution (Informal Theorem 2) shows that interactive oracle arguments with succinct query complexity can be compiled into succinct-communication arguments.

▶ **Informal Theorem 2** (Informal Statement of Corollary 11). *Any public-coin non-adaptive interactive oracle argument (with setup) with succinct (i.e. at most polylogarithmic) query complexity can be compiled into a succinct-communication argument (with setup) in the quantum random oracle model (QROM).*

Informal Theorem 2 is the bridge that will get us to Informal Theorem 1. However, we need a starting protocol with succinct query complexity to compile using the framework of Informal Theorem 2. We obtain this by modifying [ACGH20]'s classical-verifier protocol for the XZ local Hamiltonian problem by eliminating some redundant communication. The modified protocol will have succinct query complexity when the promise gap of the local Hamiltonian is at least inverse-polylogarithmic. The result of compilation using Informal Theorem 2 can be summarized as follows.

▶ **Informal Theorem 3.** *For any constant k and any relative promise gap that is at least inverse-polylogarithmic, the XZ k-local Hamiltonian problem has a classical-verifier succinct-communication argument system with non-succinct setup in the quantum random oracle model and under the post-quantum hardness assumption of LWE.*

The local Hamiltonian problem is QMA-complete when the promise gap is inverse-polynomial. The quantum PCP conjecture states that the local Hamiltonian problem remains QMA-complete

---

[2] Consider, for example, the universal gate set $G = \{H, X, \text{CCNOT}\}$. Note that $H = \frac{1}{\sqrt{2}}(X + Z)$ and $\text{CCNOT} = I - \frac{1}{4}(I - Z_1)(I - Z_2)(I - X_3)$. G is a universal gate set with real matrices and can be used to obtain propagation Hamiltonians whose Pauli decomposition has the real Pauli matrices X and Z.

when the promise gap is constant. For Informal Theorem 3 to apply to QMA (and obtain Informal Theorem 1), it suffices that the XZ local Hamiltonian problem be QMA-complete with at least inverse-polylogarithmic gap. We call this condition the *weak XZ quantum PCP conjecture.*

▶ **Conjecture 1.** *There exists a constant k such that the XZ k-local Hamiltonian problem with a promise gap that is at least inverse-polylogarithmic is* QMA*-complete.*

The qualifier "weak" here is to indicate that it is enough to amplify the gap to be inverse-polylogarithmic. When it is amplified to a constant, we call the conjecture the XZ quantum PCP conjecture.

▶ **Conjecture 2.** *There exists a constant k such that the XZ k-local Hamiltonian problem with a constant relative promise gap is* QMA*-complete.*

One can see that Conjecture 2 implies Conjecture 1 because a constant promise gap is one that is at least inverse-polylogarithmic. However, the exact relationship between either of these modified conjectures and the standard quantum PCP conjecture is unknown to us and we pose as an open problem.

▶ **Open Problem 3.** *Does the standard quantum PCP conjecture imply the (weak) XZ quantum PCP conjecture?*

1.1 Recent Related Works

Below we discuss the most relevant recent works. While most of them address the motivating problem of succinct verification of quantum computation, our work addresses also the general problem of compiling classical-verifier interactive oracle arguments into succinct arguments in the QROM. The succinct verification of quantum computation is a motivation and application of that compilation framework, but may not be the only application.

- **Succinct classical verification of quantum computation [BKL⁺22]:** Their work achieves succinct arguments for QMA (both succinct communication and succinct verification) in the standard model assuming the post-quantum security of indistinguishability obfuscation (iO) and Learning with Errors (LWE). A key contribution of that work is showing how to replace the non-succinct setup phase of the Mahadev protocol with succinct key generation based on iO. As a result, in the interactive setting, they obtain a 12-message succinct argument for QMA in the standard model, which can be reduced to 8 messages assuming post-quantum FHE; the latter protocol can be made non-interactive in the QROM.
  Our work achieves a 5-message [3] (excluding 1 offline message setup) argument in the QROM with non-succinct instance-independent setup without using FHE, but assuming a variant of the quantum PCP conjecture and LWE. In particular, our protocol makes only black-box use of cryptography and resembles practical succinct arguments for NP that compile PCPs. This makes it easier to implement in practice if a constructive proof of the (weak) XZ quantum PCP conjecture is discovered. We expect that the succinct key generation technique in [BKL⁺22] can also be applied to our protocol, which would remove the non-succinct setup at the cost of assuming and using post-quantum iO.
  Furthermore, our work addresses the general problem of compiling interactive oracle arguments into succinct arguments. The succinct verification of quantum computation is a motivation and application of this compilation framework, but may not be the only application.
- **Quantum-computational soundness of the Kilian transformation:** The soundness of the Kilian transformation from classical probabilistically checkable proofs (PCPs) against quantum polynomial-time cheating devices had been recently formally established in a line of

[3] We conjecture that it is possible to reduce the number of messages to 3 in our work. In the current version, the prover commits to one Merkle tree, then receives a Mahadev challenge (test/Hadamard), then commits to another tree, then receives the challenged indices to be revealed. This description was chosen so that Section 3 can be applied in a vanilla way. However, this choice does not utilize the fact that the challenged indices in both trees are identical! We conjecture that the verifier could send the challenged indices along with the Mahadev test/Hadamard challenge bits without exposing soundness. The intuition is that Mahadev's protocol is already a form of commitment that would be capable of replacing the second Merkle tree commitment. Furthermore, we conjecture that our protocol can be made non-interactive using the Fiat-Shamir transformation in the QROM.

works [CMS19, CMSZ21]. [CMS19] proved its soundness when the hash function is modeled via the QROM. Later, [CMSZ21] showed its soundness in the standard model when the hash function family is any *collapsing* (see [Unr16b]) hash function family. Families of such functions exist under the LWE assumption [Unr16a]. In our work, the input to the Kilian transformation is not a classical PCP, but rather a quantum PCP that was transformed into a classical-verifier interactive oracle protocol using Mahadev's verifiable measurement protocol. [CMS19] proves the soundness of SNARGs based on IOPs with round-by-round soundness in the QROM. However, in our work we do not assume any special soundness properties about the IOArgs except for standard computational soundness.

- **Classical verification of quantum computation with efficient verifier [CCY20]**: This work builds a protocol for the succinct classical verification of quantum computation with a non-succinct setup from the LWE assumption as well as post-quantum indistinguishability obfuscation (iO) and post-quantum fully homomorphic encryption (FHE). There is a gap in the soundness proof because an underlying protocol is proven sound in the QROM, but an assumption about its soundness with a concrete hash function is made. Our soundness proof is fully in the quantum random oracle model, without the need to use the code for the hash function and therefore avoiding the aforementioned gap in the soundness proof. Furthermore, our work does not require post-quantum iO nor use post-quantum FHE but rather a variant of the quantum PCP conjecture and the LWE assumption. As mentioned earlier, we also address the more general problem of compiling interactive oracle arguments.
- **zk-SNARGs for QMA from quantum null-iO [BM22]**: This work mainly studies a cryptographic concept known as *indistinguishability obfuscation for null quantum circuits (quantum null-iO)*. As an application, they achieve zero-knowledge succinct non-interactive arguments (zk-SNARGs) for QMA in the quantum random oracle model (QROM) from (i) the quantum hardness of LWE, and (ii) post-quantum indistinguishability obfuscation (iO) for classical circuits. As mentioned earlier, our work does not require post-quantum iO but rather a variant of the quantum PCP conjecture and the LWE assumption and we also address the more general problem of compiling interactive oracle arguments.
- **Online extractability in the quantum random oracle model [DFMS22b, DFMS22a]**: We make use of the online extractability framework of [DFMS22b] to prove the online extraction of Merkle trees (see Theorem 4 and Appendix C) which is implicit in their follow-up work [DFMS22a] that appeared while we were working on this paper. We kept the explicit theorem statement needed for our work and Appendix C where we prove it because the statement in our paper as well as the notation and exposition fit better with the rest of the manuscript.
- **Quantum Merkle Trees in the Quantum Haar Random Oracle Model [CM21]**: This work introduced the *Quantum Haar Random Oracle Model (QHROM)* which is a generalization to the quantum random oracle model. They show how to construct a quantum Merkle tree in this model and how it can be used to commit to and later reveal quantum states. If the quantum PCP conjecture is true, this could be used to obtain succinct arguments for QMA in the QHROM with *quantum* communication. The security is proven against what they define to be semi-honest [4] provers. In a follow-up work [CM22], they discussed zero-knowledge properties. In our work, we focus on classical verifiers (with classical communication) in the quantum random oracle model (QROM) - which is a more established model than the QHROM. We analyze security against cheating quantum provers that can perform any malicious action but limited to run in polynomial time.
- **Commitment to quantum states [GJMZ22]**: After [CM21], [GJMZ22] announced a construction of quantum Merkle trees from quantum-cryptographic assumptions (implied by one-way functions) in the standard model, and proved that the proposed succinct argument of [CM21] is secure with this instantiation (against cheating provers). This protocol is public coin and relies on very weak cryptographic assumptions, but requires quantum communication like [CM21] while our work focuses on classical verifiers with only classical communication.
- **Succinct blind quantum computation using a random oracle [Zha21]**: This work introduced a two-phase protocol for the blind delegation of quantum computation. The first

[4] This notion is different from the typical usage of the term semi-honest in cryptographic secure computation where it means an "honest but curious" adversary. A semi-honest prover in [CM21] is a prover that commits to a cheating state but follows the steps of the protocol.

phase is a quantum phase with succinct complexity while the second is entirely classical. Our work considers fully classical verifiers.

2 Background and Prior Work

Appendix A provides a glossary of the mathematical symbols and notation frequently used in this paper and Appendix B recalls some mathematical preliminaries assumed in this paper.

2.1 Merkle trees

A classical [5] *Merkle tree* of depth d is a binary tree used to commit to a sequence of blocks of data (called *leaves*) $\pi = (\pi_j)_{j\in[2^d]}$ using a cryptographic hash function $h : \mathcal{X} \to \{0,1\}^\lambda$. The *root* of the Merkle tree represents a *digest* of the blocks of the data at its leaves. For a leaf node at index $j \in [2^d]$, its *authentication path* can be used to verify its authenticity with respect to a root rt.

Figure 1 illustrates a Merkle tree of depth $d = 3$ to commit to a sequence of leaves $\pi = (\pi_1, \dots, \pi_8)$.

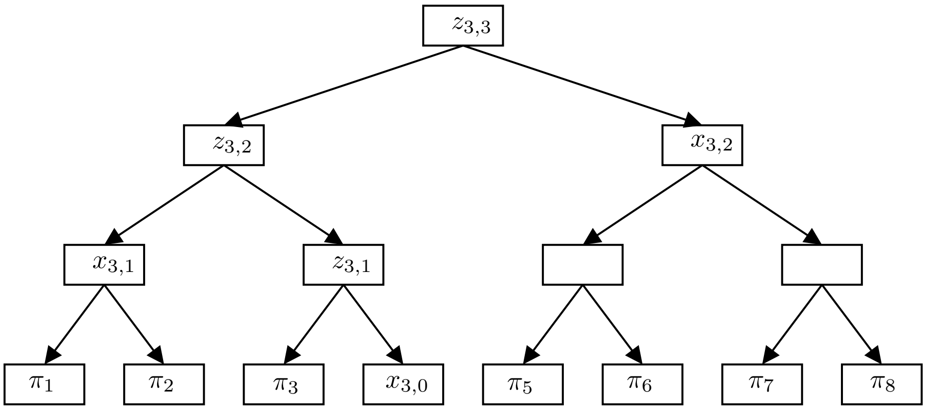


Figure 1 This figure illustrates a Merkle tree of depth $d = 3$ to commit to $2^3 = 8$ leaves with the root $rt = z_{3,3}$. The intermediate nodes for the authentication path of π_3 are marked with the notation used in this paper. Notice that $z_{3,0} = \pi_3$ and $x_{3,0} = \pi_4$ and $rt = z_{3,3}$ in a valid authentication path.

For notational convenience, let $z_{j,0} = \pi_j$. We will use the notation $h(x, x')$ to indicate applying the hash function to the proper concatenation of x and x' (respecting which is left/right child). Define $h_{j,\ell} := h(x_{j,\ell}, z_{j,\ell-1})$ where $h_{j,0} := \pi_j$. The authentication path consists of the hash values at levels $0 \le \ell \le d$ as follows: $\mathsf{ap}_j = (x_{j,\ell}, z_{j,\ell})_{0\le\ell\le d}$. An authentication path ap_j is *valid* if and only if $z_{j,d} = rt$ and $h_{j,\ell} = z_{j,\ell}$ for all $0 \le \ell \le d$. Figure 1 provides an example of a Merkle tree with 8 leaves. Let Q be a set of indices for some leaves. At each level ℓ (from 0 to d), we define the following sequence Z_ℓ which corresponds to the hash values at this level needed to verify all authentication paths: $Z_{Q,\ell} = (z_{j,\ell})_{j\in Q}$. We will use $\widehat{Z_{Q,\ell}}$ to denote the augmented sequence created from $Z_{Q,\ell}$ by ordering these intermediate Merkle tree nodes from left to right and replacing any missing nodes with $\perp$. When Q is clear in the context, we write $Z_{Q,\ell}$ as Z_ℓ and $\widehat{Z_{Q,\ell}}$ as $\widehat{Z_\ell}$ for brevity. Similarly, we define: $X_{Q,\ell} = (x_{j,\ell})_{j\in Q}$ and $\widehat{X_{Q,\ell}}$ as well as their shorted notations X_ℓ and $\widehat{X_\ell}$ respectively when Q is clear in the context. The suite of Merkle tree algorithms used in this paper are as follows:

- $\textsc{Commit}^h(\pi_1, \dots, \pi_{2^d})$: returns the root of the Merkle tree rt and all intermediate nodes,
- $\textsc{Valid}^h(rt, j, \mathsf{ap}_j)$: returns true if and only if the given authentication path ap_j for the j-th leaf is valid against the root rt by using the hash function h,
- $\textsc{Consistent}(Q, \{\mathsf{ap}_j\}_{j\in Q})$: returns true if and only if the authentication paths for leaves at indices $Q \subseteq [2^d]$ are well-formed and *consistent* at the common intermediate nodes [6], and

[5] In this paper we will only work with classical Merkle trees where the data are classical strings and the algorithms are executed on classical devices. However, their security is established against a cheating quantum device in the quantum random oracle model.

[6] This is equivalent to sending each overlapping intermediate node once instead of sending it multiple times inside possibly overlapping paths for each leaf. However, for easier notation and exposition, we send the authentication paths for each leaf and require this consistency condition when verifying a batch of authentication paths.

- $\textsc{Verify}^h(rt, Q, \mathsf{ap}_{j\in Q})$: validates a batch of authentication paths and returns true if and only if both $\textsc{Consistent}(Q, \mathsf{ap}_{j\in Q})$ and $\forall j \in Q : \textsc{Valid}^h\left(rt, j, \mathsf{ap}_j\right)$ are true.

2.2 Merkle Trees in the Quantum Random Oracle Model (QROM)

The *random oracle* [BR93] models a concrete cryptographic hash function $H : \mathcal{X} \to \mathcal{Y}$ as an external random oracle RO that answers queries randomly the first time they are submitted and consistently whenever they are resubmitted. Precisely, the random oracle is a uniformly random function from $\mathcal{X}$ to $\mathcal{Y}$. The quantum random oracle [BDF+11] is a unitary oracle $U_H : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$ defined with an underlying uniformly random function H. The query is submitted in the x register and an answer $H(x)$ is returned by XORing such answer with the content of the y register.

Since the introduction of the QROM, different techniques and applications were introduced, most notably the *compressed oracle* technique due to Zhandry [Zha19]. Building on the success of this line of work, [DFMS22b] introduced a framework for *online extractability* in the quantum random oracle model. Online extraction means that the extraction happens (i) *on-the-fly* during the algorithm's execution, and (ii) in a *straightline* which means that no rewinding of the algorithm calling the random oracle is needed. [DFMS22b] provides a framework that encapsulates many of the inner workings that needed to be handled extensively before. Their framework offers an *extractable* random oracle simulator $\mathcal{S}$ which has an internal database state and two query interfaces (which are operators) (see Figure 3 in Appendix C):

1. $\mathcal{S}$.RO-query: the quantum random oracle unitary, and
2. $\mathcal{S}.E$-query: a classical extraction query that applies a measurement to the simulator state.

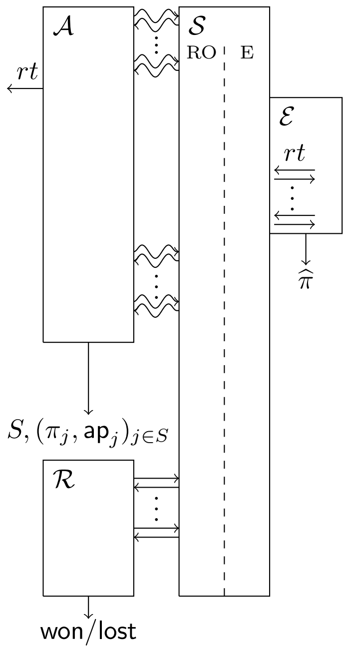


Figure 2 This figure illustrates the game G_1 referenced in Theorem 4. $\mathcal{A}$ wins if $S \subseteq [2^d]$, $|S| = r$, and $\textsc{Verify}^{\mathrm{RO}}\left(rt, S, \mathsf{ap}_{j\in S}\right)$, but $\exists j \in S : \pi_j \neq \widehat{\pi}_j$. The "snaked" arrowed lines represent *quantum* queries and responses thereof, while the straight arrowed lines represent *classical* queries and responses thereof. The referee $\mathcal{R}$ consists of two main procedures: (1) verifying the authentication paths which needs to interact with the $\mathcal{S}$.RO interface, and (2) comparing the output of the adversary and the extractor which does not interact with $\mathcal{S}$.

We will use the following result about the online extraction of Merkle trees which is implicit in a follow-up work by [DFMS22a], but we also provide a detailed discussion and a proof for it in Appendix C which was written prior to the publication of [DFMS22a]. The theorem bounds the probability of winning a game $G_1(\lambda, d, r, q)$ illustrated in Figure 2 (as well as Figure 4 in Appendix C) where a quantum adversary $\mathcal{A}$ interacts with only the RO interface while a classical honest extraction algorithm $\mathcal{E}$ only (classically) interacts with the E interface of the simulated random oracle. The adversary announces a classical value rt which is *supposedly* the root of a Merkle tree of depth d and they win if they can later *"fake"* at least one of r leaves. Faking a leaf here means giving a leaf value that can be authenticated against the prior commitment, but different from that output by extraction. A referee algorithm $\mathcal{R}$ determines whether the adversary won by validating the authentication paths against the root rt then comparing the adversary's leaves against the leaves given by the extraction algorithm.

▶ **Theorem 4.** *For the game G_1 defined in Figure 4 by the universal referee and extractor algorithms described earlier such that $\lambda = \omega(d)$, $q \leq \mathrm{poly}(2^d)$, and any quantum adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ where $\mathcal{A}_1$ makes q_1 queries to the random oracle, then $\mathcal{A}_1$ announces a value rt, followed by $\mathcal{A}_2$ making q_2 queries to the random oracle such that $q_1 + q_2 \leq q$, then $\mathcal{A}_2$ outputs a classical string, it holds that:*

$$\Pr[\mathcal{A} \textit{ wins } G_1(\lambda, d, r, q)] \leq \mathrm{negl}(\lambda).$$

2.3 The Local Hamiltonian Problem

▶ **Definition 5** (**L**ocal **H**amiltonian Problem (n,k,γ)-LH)**.** *The k-local Hamiltonian problem notated as (n,k,γ)-LH is a promise problem where the input is a classical binary string $x = (H,a,b)$ such that:*

- *H is a k-local Hamiltonian $H = \sum_{s=1}^{S} H_s$ on a total of n qubits where $S = \mathrm{poly}(n)$ and each H_s is a Hermitian matrix with a bounded operator norm $||H_s|| \leq 1$ and its entries are specified by $\mathrm{poly}(n)$ bits and H_s is non-identity on at most k qubits,*
- *a and b are two numbers represented with $\mathrm{poly}(n)$ bits such that $a < b$; the gap $\Gamma = b - a$ is called the **absolute promise gap** and $\gamma = \Gamma/S$ is called the **relative promise gap**,*
- *for yes-instances, there exists an n-qubit quantum state $|\psi\rangle$ such that $\langle\psi| H |\psi\rangle \leq a$ (i.e. energy of the state w.r.t. H is at most a),*
- *for no-instances, for every n-qubit quantum state $|\psi\rangle$, it holds that $\langle\psi| H |\psi\rangle \geq b$ (i.e. energy of the state w.r.t. H is at least b), and*
- *it is promised that any instance will be either a yes or no instance.*

The problem is called the XZ k-local Hamiltonian problem and we notate it as (n,k,γ)-LH-XZ when each H_s is a constant-scaled tensor product of n matrices from the set of 2×2 matrices $\{\mathbb{1}, X, Z\}$ such that at most k of the matrices in each product are non-identity.

This problem is QMA-complete when the promise gap is at least inverse polynomial i.e. $\gamma \geq 1/\mathrm{poly}(n)$. The k-LH problem remaining QMA-hard even when this promise gap is constant i.e. $\gamma \geq \alpha$ for some constant α is known as the *quantum PCP conjecture* (qPCP for brevity), which is still unsettled to date. [AALV09] showed that the qPCP statement is equivalent to obtaining PCPs for QMA where quantum reductions [7] are used to prove that the proof verification version implies the gap amplification version.

2.4 Classical-Verifier Argument for XZ Local Hamiltonians

We will now describe Protocol 6 due to [ACGH20] which is a quantum-prover classical-verifier argument system with an instance-independent setup phase. The protocol can be parallel-repeated to obtain negligible completeness and soundness errors. In Appendix E, we give a detailed exposition and proofs of completeness and soundness and explain the modular construction of this protocol while generalizing the locality to any constant k and the promise gap to any function. We give below a very brief summary.

Protocol 6 [ACGH20] uses Mahadev's verifiable measurement protocol described in Section E.2 to make the verifier of a protocol for local Hamiltonian verification (Protocol 28) classical instead of quantum. In the predecessor version of Protocol 28 [MF16, FHM18, MNS16], the choice of measurements (X or Z) depended on the choice of the Hamiltonian term. This is because a particular Hamiltonian term may act by X on a qubit while another term could act by Z on the

[7] It is an open question whether they are equivalent under classical reductions. In fact, the proof checking formulation itself could end up being more specific than that provided in [AALV09] which was the reason why it was not straightforward to prove the equivalence under classical reductions. For the details of the quantum reduction, we refer the reader to the proof of Theorem 5.5. in [Gri18].

same qubit. This poses a challenge when using Mahadev's verifiable measurement because the first step of Mahadev's protocol samples keys that depend on the basis choice. [ACGH20] got around this issue by randomly sampling a basis for each qubit. When the time comes to select a Hamiltonian term, the verifier first checks whether this selected term is consistent with the randomly selected bases on the affected qubits.

In the first round of [ACGH20]'s protocol, the verifier generates a set of private trapdoors and corresponding public keys (a trapdoor/key for each qubit in the witness state) to initiate the Mahadev protocol. The prover then sends a commitment for the witness state - they allegedly have - using the received public keys. The verifier then sends a challenge bit (0/1) that dictates certain measurements to be done by the prover. The prover measures accordingly and sends the measurement outcomes. If the verifier sent 0 as the challenge bit, a Mahadev *"test round"* (TestCheck) is executed whose purpose is making sure that the prover "did not change their mind" after the commitment. If the verifier sent 1 as the challenge bit, a Mahadev *"Hadamard round"* (HadRound) is executed to extract the measurements needed to execute the verification procedure on the Hamiltonian term. The protocol is executed multiple times in parallel using multiple copies of the witness state.

▶ **Protocol 6** (Protocol 4 in [ACGH20]; Quantum-Prover Classical-Verifier Argument System for XZ local Hamiltonians with Instance-Independent Setup)**.**

Parties: *Quantum polynomial-time prover $\mathcal{P}$ & classical probabilistic polynomial-time verifier $\mathcal{V}$.*

Parameters:
1. *n: number of qubits.*
2. *r, m: number of repetitions in the LH verification and Mahadev protocols respectively.*
3. *λ: a security parameter $\geq n$.*

Setup:
1. *$\mathcal{V}$ samples uniformly random bases $h \in \{0,1\}^{nrm}$.*
2. *$\mathcal{V}$ runs Mahadev's key generation algorithm $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda, h)$.*
3. *$\mathcal{V}$ sends the public keys pk to $\mathcal{P}$, but maintains sk secretly [8].*

Inputs: ***Inputs to both parties:*** *$x = (H = \sum_{s=1}^{S} d_s H_s, a, b)$ i.e. instance of the (n,k,γ)-*LH-XZ*.*

Input to honest prover on yes instances: *$|\Psi\rangle = |\psi\rangle^{\otimes rm}$ (i.e. rm copies of $|\psi\rangle$ the ground state of the Hamiltonian H). This state is in the register W. For each $i \in [m], \ell \in [r], j \in [n]$, we use $W_{i\ell j}$ to denote the corresponding qubit.*

Round $\mathcal{P}_1$: *For each $W_{i\ell j}$, the prover prepares the "commitment" state (see Section E.2; we use here F to denote f or g depending on the uniformly chosen basis):*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}} \sum_{b\in\{0,1\}} \phi_b \, |b\rangle_{W_{i\ell j}} |x\rangle_{X_{i\ell j}} |F_{pk_{i\ell j}}(b,x)\rangle_{Y_{i\ell j}} .$$

$\mathcal{P}$ measures each register $Y_{i\ell j}$ in the standard basis & sends the outcomes $y = (y_{i\ell j})$ to $\mathcal{V}$.

Round $\mathcal{V}_2$: *$\mathcal{V}$ samples challenge bits $c_1, \ldots, c_m \leftarrow \{0,1\}$ and sends $c = (c_1, \ldots, c_m)$ to $\mathcal{P}$. 0 or 1 means asking the prover to engage in test or Hadamard rounds (respectively) of the Mahadev protocol (see Section E.2).*

Round $\mathcal{P}_2$: *For each $i \in [m], \ell \in [r], j \in [n]$,*
1. *If $c_i = 0$, $\mathcal{P}$ performs a standard basis measurement and gets $u_{i\ell j} = (w_{i\ell j}, x_{i\ell j})$.*
2. *If $c_i = 1$, $\mathcal{P}$ performs a Hadamard basis measurement and gets $u_{i\ell j} = (w_{i\ell j}, x_{i\ell j})$.*

$\mathcal{P}$ sends $u = (u_{i\ell j})$ to $\mathcal{V}$.

$\mathcal{V}$'s Verdict *For each $i \in [m]$,*
1. *If $c_i = 0$, $\mathcal{V}$ sets $v_i := \bigwedge_{\ell,j} \mathsf{TestCheck}(sk_{i\ell j}, u_{i\ell j}, y_{i\ell j})$ (see Section E.2).*
2. *If $c_i = 1$, $\mathcal{V}$ records the set $A_i \subseteq [r]$ (the subset of copies consistent with the random bases choice). For each $\ell \in A_i$:*
   a. *Run the Hadamard round (see Section E.2) for each $j \in [n]$:*

   $$(z_{i\ell j}, e_{i\ell j}) := \mathsf{HadRound}(sk_{i\ell j}, u_{i\ell j}, y_{i\ell j}, h_{i\ell j}).$$

   If it rejects (i.e. $z_{i\ell j} = 0$ for some j), set $v_{i\ell} = 0$; otherwise enter the next step.

[8] Later, we will use the term "public-coin protocols with private setup" to highlight this again.

b. *Like in Protocol 28, sample a Hamiltonian term $s_{i\ell} \leftarrow \pi$ where the distribution π is given by:*

$$\pi(s) = \frac{|d_s|}{\sum\limits_s |d_s|}.$$

Denote by $\mathcal{S}(i,\ell)$ the set of indices of the qubits acted upon by non-identity Pauli observables. Set $v_{i\ell} := \frac{1}{2}\left(1 - \textsc{sgn}(d_{s_{i\ell}}) \cdot \prod\limits_{j \in \mathcal{S}(i,\ell)} e_{i\ell j}\right)$ (i.e. set to 1 iff the measurement has the opposite sign of the coefficient of the selected term).

Then, as in Step 3 of the verdict in Protocol 28: $\mathcal{V}$ sets $v_i = 1$ iff:

$$\sum_{\ell \in A_i} v_{i\ell} \geq \frac{(c+s)}{2} \cdot |A_i| = \frac{\left(2 - (b-a)/\sum\limits_s |d_s|\right)}{4} \cdot |A_i|$$

where (see Protocol 28 and the proof in Appendix F for the details):

$$c := \frac{1}{2} - \frac{a}{2\sum\limits_s |d_s|} \qquad \text{and} \qquad s := \frac{1}{2} - \frac{b}{2\sum\limits_s |d_s|}.$$

Finally, $\mathcal{V}$ accepts iff $v := \bigwedge_{i=1}^m v_i$ evaluates to 1 (i.e. v_i is 1 for each parallel repetition $i \in [m]$).

3 Succinct Communication from Interactive Oracle Arguments

3.1 Defining Interactive Oracle Arguments

We now formalize the notion of *quantum-computationally sound classical-verifier interactive oracle proofs* for quantum-witness relations (which for brevity we also call IOArgs for *interactive oracle arguments*) by generalizing interactive oracle proofs (IOPs) in [BCS16]. In particular, we introduce IOArgs with a pre-processing (setup) phase where the verifier sends a message to the prover that does not depend on the input instance but only on an upper bound on the instance size n. Since this step does not need the input and can happen temporally before the execution of the protocol on a particular input, we do not account for its cost when analyzing succinctness of the protocol communication.

▶ **Definition 7** (Interactive Oracle Arguments with Setup; Generalizing Interactive Oracle Proofs in [BCS16]). *Let $p(n)$ be a polynomial and $\mathcal{R}$ be a relation: $\mathcal{R} \subseteq \bigcup\limits_{n=0}^{\infty} \{0,1\}^n \times \mathcal{H}_{p(n)}$ where $\mathcal{H}_{p(n)}$ is the Hilbert space of $p(n)$-qubit pure quantum states. Consider a promise problem $A = (A_{yes}, A_{no})$ where $A_{yes} \cap A_{no} = \emptyset$ and $A_{yes} := \{x \mid \exists |\psi\rangle : (x, |\psi\rangle) \in \mathcal{R}\}$. We say that A has a quantum-computationally sound classical-verifier interactive oracle proof system with setup with the following parameters (notated as $A \in \mathrm{IOA}\textsc{rg}_{c,s}[t(n), \ell(n), r(n), q(n)]$):*

- *round complexity $t(n)$: number of prover oracle messages in the protocol,*
- *total length of all prover messages: $\ell(n)$,*
- *randomness complexity $r(n)$: total number of random bits used by the verifier,*
- *query complexity $q(n)$: number of queries by the verifier to the prover's oracle messages,*
- *completeness $c(n)$, and soundness $s(n)$*

if there is an interactive protocol between:

Parties: 1. *$\mathcal{P}^{|\psi\rangle}$: a quantum $\mathrm{poly}(n)$-time algorithm (when the input x is a yes instance, an honest prover will receive a state $|\psi\rangle$ such that $(x, |\psi\rangle) \in \mathcal{R}$), and*

2. *$\mathcal{V} = (\mathcal{V}_0, \ldots, \mathcal{V}_{t(n)})$: a classical probabilistic $\mathrm{poly}(n)$-time algorithm using $r(n)$ random bits. The verifier's sub-algorithm $\mathcal{V}_0 = \textsc{Setup}(1^n)$ is an optional setup phase that only depends on the input length [9] but not the input itself while the the other sub-algorithms $\mathcal{V}_1, \ldots, \mathcal{V}_{t(n)}$ depend on the input x.*

[9] In most useful interactive oracle arguments including the argument system for the local Hamiltonian problem discussed in this paper, we do not have to know the input length exactly, but it suffices to know an upper bound.

Setup: *The protocol starts with an optional setup phase run by the verifier* $(p_0, v_0) \leftarrow \textsc{Setup}(1^n)$*. The verifier sends* p_0 *to the prover and* ***keeps***[10] v_0.

Interaction: *For any round* $i \in [t(n)]$*, the following interaction takes place:*

1. *The prover sends an oracle message* $p_i = \mathcal{P}(x, p_0, p_1, \ldots, p_{i-1}, v_1, \ldots, v_{i-1})$.
2. *If* $i < t(n)$*, the verifier samples randomness* $\$_i$ *and outputs a message* $v_i = \mathcal{V}(x, v_0, v_1, \ldots v_{i-1}; \$_i)$.

Verdict: *At the end of the protocol, the verifier samples randomness* $\$_{t(n)}$ *and chooses* $q(n)$ *locations* $Q = (Q_1, \ldots, Q_{t(n)})$ *to access from previous prover oracle messages* $p_1, \ldots, p_k$*. Finally, the verifier runs a predicate* $\textsc{Verdict}(x, p_{1|Q_1}, \ldots, p_{t(n)|Q_{t(n)}}, v_0, v_1, \ldots, v_{t(n)-1}; \$_{t(n)})$ *to output a decision (accept/reject).*

Completeness: *If* x *is a yes-instance, with* $|x| = n$*, then for an honest prover* $\mathcal{P}$ *receiving a quantum state* $|\psi\rangle$ *such that* $(x, |\psi\rangle) \in \mathcal{R}$*:* $\Pr[\langle \mathcal{P}, \mathcal{V}\rangle \text{ accepts } x] \geq c(n)$.

Soundness: *If* x *is a no-instance, with* $|x| = n$*, then for any quantum polynomial-time interactive algorithm* $\widetilde{\mathcal{P}}$*:* $\Pr[\langle \widetilde{\mathcal{P}}, \mathcal{V}\rangle \text{ accepts } x] \leq s(n)$.

We say that an IOArg is ***public-coin with private setup*** *if the verifier sends the randomness they generate to the prover* [11] *(except for the randomness used in the setup step). In our definition, the queries of the IOArg are* ***non-adaptive*** *in the sense that one query does not depend on the answer to another. In this paper, we work with non-adaptive public-coin IOArgs with private setup.*

3.2 Succinct Communication by Applying the Kilian Transformation

We now show how to apply the standard Kilian transformation [Kil92] to compile any non-adaptive public-coin IOArg with private setup and succinct query complexity into a succinct-communication argument. To prove the soundness of the compiled protocol, we will use the online extraction of Merkle trees in the quantum random oracle model discussed in Section C.

▶ **Protocol 8** (Succinct-communication argument from non-adaptive public-coin IOArg with private setup and succinct query complexity)**.**

Model: $RO : \mathcal{X} \to \{0,1\}^\lambda$ *is a quantum random oracle which could be called in superposition.*

Promise Problem: $A \in \textsc{IOArg}_{c,s}[t(n), \ell(n), r(n), q(n)]$ *with an underlying relation* $\mathcal{R}$ *where* $q(n) = \widetilde{O}(\lambda)$.

Parties: *Quantum poly-time prover* $\mathcal{P}$ *& classical probabilistic poly-time verifier* $\mathcal{V}$.

Setup: *The verifier runs* $(p_0, v_0) \leftarrow \textsc{Setup}(1^n)$ *from the underlying IOArg, keeps* v_0*, and sends* p_0 *to the prover.*

Inputs: ***To both parties:*** x *where* $|x| = n$ *&* x *is a yes/no instance of the promise problem* A.
To the prover: *The setup message* p_0 *received during the setup. An honest prover will also receive a state* $|\psi\rangle$ *on yes-instances* x *such that* $(x, |\psi\rangle) \in \mathcal{R}$.

Round $\mathcal{P}_\mathbf{i}$***:*** *The prover computes the message* p_i *according to the underlying IOArg. The prover then uses* $\textsc{Commit}^{RO}$ *to compute a Merkle tree root* rt_i *for the message* p_i *and sends* rt_i *to the verifier.*

Round $\mathcal{V}_\mathbf{i}$***:*** *If* $i < t(n)$*: according to the underlying IOArg the verifier samples randomness* $\$_i$ *and sends the message* v_i.
If $i = t(n)$*: According to the underlying IOArg, the verifier samples randomness* $\$_{t(n)}$ *and determines the* $q(n)$ *locations* $Q = (Q_1, \ldots, Q_{t(n)})$ *to access from the previous prover oracle messages* $p_1, \ldots, p_{t(n)}$ *that were supposedly committed with the roots* $rt_1, \ldots, rt_{t(n)}$ *respectively. The verifier sends these indices* Q *to the prover.*

Round $\mathcal{P}_{\mathbf{t+1}}$***:*** *The prover sends the* $q(n)$ *bits at locations* Q *along with authentication paths to the verifier i.e. they send the sequence* $\left((\pi_{i,j}, \mathsf{ap}_{i,j})_{j\in Q_i}\right)_{1\leq i\leq t(n)}$ *where* $\mathsf{ap}_{i,j}$ *means the authentication path of the* j*th location with respect to the root* rt_i *of the* i*th Merkle tree.*

Verdict*:* *For each* $i = 1 \ldots t(n)$*, the verifier verifies the authentication paths with access to the random oracle RO and using the predicate* $\textsc{Verify}$ *defined in Section 2.1. Precisely, in the* i*th iteration, the verifier performs this verification by calling* $\textsc{Verify}^{RO}\left(rt_i, Q_i, (\mathsf{ap}_{i,j})_{j\in Q_i}\right)$*. It rejects if this predicate rejects. Otherwise, the verifier outputs the output of:*

$$\textsc{Verdict}(x, \pi_{1|Q_1}, \ldots, \pi_{t|Q_t}, v_0, v_1, \ldots, v_{t-1}; \$_{t(n)})$$

[10] Keeping the randomness used in the setup enables the verifier to store information such as secret keys and/or trapdoors without revealing them to the prover.

[11] or its oracle messages

where $\textsc{Verdict}$ *is the verdict predicate of the underlying IOArg and* $\pi_{i|Q_i}$ *are the locations received from the prover during the round* $\mathcal{P}_{t(n)+1}$*.*

3.3 Analysis of the Compiled Protocol

The completeness of Protocol 8 is stated in Theorem 9 and proven in Appendix D.1 using the idempotence property of the RO interface (Property 4, Theorem 21). The soundness of this protocol is summarized in Theorem 10 and proven in Appendix D.2 which are key technical contributions in this paper. In Appendix D.3, we analyze the total communication cost in this protocol which is found to be $O\left(\lambda \cdot (t(n) + q(n) \cdot \log(n)) + r(n)\right)$ classical bits. The resulting protocol is succinct when $q(n) = O(\mathrm{poly}(\log(n))) = \tilde{O}(1)$, $r(n) = \tilde{O}(1)$, $t(n) = \tilde{O}(1)$, and $\ell(n) = \mathrm{poly}(n)$. Finally, we summarize these three properties of the protocol (completeness, soundness, and succinctness) in Corollary 11.

▶ **Theorem 9** (Completeness of Protocol 8)**.** *For a promise problem* $A \in \textsc{IOArg}_{c,s}[t(n), \ell(n), r(n), q(n)]$ *such that* $c(n)$ *is the completeness of the IOArg, Protocol 8 built on that IOArg also has completeness* $c(n)$*.*

▶ **Theorem 10** (Computational Soundness of Protocol 8)**.** *Consider a promise problem* A *with an interactive oracle argument i.e.* $A \in \textsc{IOArg}_{c,s}[t(n), \ell(n), r(n), q(n)]$*. Let Protocol 8 be built on top of this IOArg in the quantum random oracle model with* $\lambda = \omega(\log(\ell(n)))$*. Let* x *be an instance of* A *with* $n = |x|$*. If a (possibly cheating) quantum prover* $\mathcal{P}$ *running in polynomial time* $T_{\mathcal{P}}(n) = \mathrm{poly}(n)$ *and access to RO can make an honest verifier* $\mathcal{V}$ *in such protocol accept* x *with probability* $\geq \delta(n)$*, then there exists a polynomial-time (quantum) IOArg prover* $\widetilde{P}_{\textsc{IOArg}}(x)$ *that can make an honest IOArg verifier accept* x *with probability* $\geq \delta(n) - \mathrm{negl}(\lambda)$*.*

▶ **Corollary 11** (Succinct-Communication Arguments from IOArgs)**.** *In the quantum random oracle model with* $RO : \mathcal{X} \to \{0,1\}^\lambda$ *and* $\lambda = \omega(\log(n))$*: Protocol 8 built for a promise problem* $A \in \textsc{IOArg}_{c,s}[\widetilde{O}(1), \mathrm{poly}(n), \widetilde{O}(1), \widetilde{O}(1)]$ *is a succinct-communication argument with (possibly non-succinct) setup with completeness* c *and soundness* $s - \mathrm{negl}(\lambda)$*.*

4 Classical-Verifier Succinct-Communication Argument for XZ Local Hamiltonians

4.1 Eliminating redundancy in [ACGH20]'s classical-verifier argument

Protocol 12 is a modified version of Protocol 6. When executing the Mahadev verifiable measurement test/Hadamard rounds in the protocol, we only verify the measurements for the qubits that would have been necessary to run the LH verification. Precisely, the difference here is that - even in Mahadev's test round - the index j ranges over the set $\mathcal{S}(i, \ell)$ which is the set of qubit indices affected by non-identity observables in the Hamiltonian term $s_{i\ell}$ instead of ranging over $[n]$ (i.e. all qubits).

▶ **Protocol 12** (Modified version of Protocol 6 after eliminating redundancy)**.**

Parties, Inputs, Setup*: Same as in Protocol 6.*

Rounds $\mathcal{P}_1, \mathcal{V}_2, \mathcal{P}_2$*: Same as in Protocol 6.*

$\mathcal{V}$***'s Verdict*** *For each* $i \in [m], \ell \in [r]$ *:* $\mathcal{V}$ *samples a Hamiltonian terms* $s_{i\ell} \leftarrow \pi$ *where the distribution* π *is given by:*

$$\pi(s) = \frac{|d_s|}{\sum\limits_s |d_s|}.$$

Denote by $\mathcal{S}(i, \ell)$ *the set of indices of the qubits acted upon by non-identity Pauli observables.*

Also, let $A_i \subseteq [r]$ *be the subset of copies consistent with the random bases choice.*

For each $i \in [m]$*:*

1. *If* $c_i = 0$ *(test round), set* $v_i := \bigwedge\limits_{\ell \in A_i,\ j \in \mathcal{S}(i,\ell)} \mathsf{TestCheck}(sk_{i\ell j}, u_{i\ell j}, y_{i\ell j})$*.*
2. *If* $c_i = 1$ *(Hadamard round), for each* $\ell \in A_i$*:*

a. *Run the Hadamard round for each* $j \in \mathcal{S}(i,\ell)$*:*

$$(z_{i\ell j}, e_{i\ell j}) := \mathit{HadRound}(sk_{i\ell j}, u_{i\ell j}, y_{i\ell j}, h_{i\ell j}).$$

If it rejects (i.e. $z_{i\ell j} = 0$ *for some* j*), set* $v_{i\ell} = 0$*; otherwise enter the next step.*

b. *Set* $v_{i\ell} := \frac{1}{2}\left(1 - \mathrm{SGN}(d_{s_{i\ell}}) \cdot \prod_{j \in \mathcal{S}(i,\ell)} e_{i\ell j}\right)$ *(i.e. set to 1 iff the measurement has the opposite sign of the coefficient of the selected term).*

Then, as in Protocols 28 and 6: $\mathcal{V}$ *sets* $v_i = 1$ *iff:*

$$\sum_{\ell \in A_i} v_{i\ell} \geq \frac{(c+s)}{2} \cdot |A_i| = \frac{\left(2 - (b-a)/\sum_s |d_s|\right)}{4} \cdot |A_i|$$

where (see Protocol 28 and the proof in Appendix F for the details):

$$c := \frac{1}{2} - \frac{a}{2\sum_s |d_s|} \qquad \text{and} \qquad s := \frac{1}{2} - \frac{b}{2\sum_s |d_s|}.$$

Finally, as in Protocol 6, $\mathcal{V}$ *accepts iff* $v := \bigwedge_{i=1}^m v_i$ *evaluates to 1 (i.e.* v_i *is 1 for each parallel repetition* $i \in [m]$*).*

In Appendix F, we follow [ACGH20]'s proof of the soundness of Protocol 6 to show how the soundness of this modified protocol still holds even when we only verify the Mahadev measurements for the qubits affected by the selected local Hamiltonian term. We outline a corollary to that result below.

▶ **Corollary 13** (Mirror of Theorem 4.6. in [ACGH20]). *Under the LWE assumption, for every constant* k*, Protocol 12 with* $r = \omega(\frac{\log(n)}{\gamma^2})$ *and* $m = \omega(\log(n))$ *has negligible completeness and soundness errors.*

4.2 Compiling towards Succinct Communication

Since only a number of selected locations are read from each prover message, we can rewrite Protocol 12 as an IOArg by modeling the prover messages as message oracles instead of message strings. As a result, we get Protocol 14 which is a two-round public-coin non-adaptive interactive oracle argument with a private setup. Specifically, the verifier's choices with the exception of key-generation - which happens in setup - are revealed to the prover (or its message oracles). Note that the setup phase is non-succinct because the verifier needs to send a key for each qubit. The verifier sends a total of m (the number of parallel repetitions of the Mahadev protocol) classical bits in the first round. The verifier needs to query $k \cdot r \cdot m$ locations from each prover oracle. Theorem 35 and Corollary 13 still directly apply to this protocol because it is exactly the same as Protocol 12 from the point of view of both the prover and verifier. When γ is at least inverse polylogarithmic, one can take $r = \omega(\log n/\gamma^2)$ to obtain negligible completeness and soundness errors in Protocol 14 as well as polylogarithmic query complexity. We can then apply Corollary 11 to conclude with Corollary 15.

▶ **Protocol 14** (Interactive Oracle Argument with Preprocessing for XZ Local Hamiltonians).

Parties, Inputs, Setup: *Same as in Protocol 12.*

Round $\mathcal{P}_1$***:*** $\mathcal{P}$ *follows the steps of Protocol 12 (as described in Protocol 6) and sends an oracle* $\mathcal{O}_y$ *that represents the measurement outcomes on the commitment qubits.*

Round $\mathcal{V}_1$***:*** $\mathcal{V}$ *samples* $c_1, \ldots, c_m \leftarrow \{0,1\}$ *and sends* $c = (c_1, \ldots, c_m)$ *to* $\mathcal{P}$*.*

Round $\mathcal{P}_2$***:*** $\mathcal{P}$ *follows the steps of Protocol 12 and sends an oracle* $\mathcal{O}_u$ *to* $\mathcal{V}$ *that represents the measurement outcomes of measuring the pre-image and committed qubit registers.*

Round $\mathcal{V}_2$***:*** $\mathcal{V}$ *samples terms* $s_1, \ldots, s_{rm} \leftarrow \pi$ *and queries their corresponding indices from the oracles* $\mathcal{O}_y$ *and* $\mathcal{O}_u$*.*

$\mathcal{V}$***'s Verdict:*** $\mathcal{V}$ *executes and returns the output of the verdict round of Protocol 12.*

▶ **Corollary 15.** *Under the post-quantum hardness of LWE and for any natural number* n*, there exists a classical-verifier succinct-communication argument system with instance-independent setup and negligible completeness and soundness errors for instances of size at most* n *of the* (n,k,γ)-LH-XZ *problem with at least inverse-polylogarithmic relative promise gap in the quantum random oracle model with* $RO : \mathcal{X} \rightarrow \{0,1\}^\lambda$ *and any* $\lambda = \omega(\log(n))$*.*

4.3 XZ Quantum PCP Conjecture and Consequences to QMA

We now formally state the *weak XZ quantum PCP conjecture (Conjecture 16)* which was defined informally in Informal Conjecture 1.

▶ **Conjecture 16** (Weak XZ Quantum PCP Conjecture). *There exist a constant k and a function $f(n) = \widetilde{O}(1)$ such that the (n, k, γ)-LH-XZ problem with relative promise gap $\gamma(n) = 1/f(n)$ is* QMA*-hard.*

The (weak) XZ quantum PCP conjecture (Conjecture 16) and Corollary 15 imply the existence of succinct-communication arguments with setup for QMA under the LWE assumption in the QROM which can be stated as follows.

▶ **Theorem 17.** *If the Weak XZ Quantum PCP Conjecture (Conjecture 16) is true as well as the post-quantum hardness of LWE, then for any promise problem $A \in$ QMA and any natural number n, there exists a succinct-communication argument system with setup for all instances of A of size at most n in the quantum random oracle model with $RO : \mathcal{X} \to \{0,1\}^{\lambda}$ and any $\lambda = \omega(\log(n))$.*

While we could not prove that Conjecture 16 is implied by the standard quantum PCP conjecture, we conjecture that this would be possible via a gap-preserving reduction. The tools to prove an implication like that may come to light when more progress is made towards settling the standard quantum PCP conjecture. Actually, it might be the case that a long-awaited proof of the quantum PCP conjecture would be established via the QMA-hardness of XZ local Hamiltonians.

5 Conclusion

We formalized the notion of post-quantum interactive oracle arguments (with setup). Given that formalism, we showed a framework to compile any public-coin non-adaptive interactive oracle argument (with private setup) into a succinct-communication argument (with possibly non-succinct setup). Our soundness proof utilized the online extraction of Merkle trees in the quantum random oracle model. We stated the (weak) XZ quantum PCP conjectures as variants of the standard quantum PCP conjectures. In the QROM, either of these conjectures is sufficient to imply the existence of succinct-communication classical-verifier arguments with non-succinct setup for QMA under the LWE assumptions (and consequently a protocol for succinct-communication classical verification of quantum computation with non-succinct setup).

Appendices

We provided in the appendices enough materials to make the paper self-contained. Appendix D expands on Section 3.3 and is an original contribution in this paper. Appendix C expands on Section 2.2 and proves a result implicit in [DFMS22a]. The concrete statement and proof we provide in Section C fit the exposition of other sections in this paper and were written prior to the publication of [DFMS22a].

A Glossary

Table 1 provides a glossary of most of the symbols and notation used in this paper. While we borrow a lot of [ACGH20]'s exposition style in introducing the classical-verifier argument system for local Hamiltonians, we slightly diverge from their symbolic notation as indicated in that table.

B Mathematical preliminaries

We recall some of the definitions and facts frequently used later in the paper. Let p and q be two classical probability distributions on a finite sample space Ω. The *total variation distance* between p and q is

$$d_{\mathrm{TV}}(p, q) = \frac{1}{2}\sum_{x\in\Omega}|p(x) - q(x)| = \max_{A\subseteq\Omega}|p(A) - q(A)|.$$

Symbol/Notation	Description	Symbol in [ACGH20]
n	Number of qubits in a single copy of a quantum state	n
j	Index over qubits in a single copy of a quantum state	j
H	k-local Hamiltonian on n qubits used once to denote the Hadamard gate	H
k	Locality of a Hamiltonian	k
S	Number of Hamiltonian terms	
s	Index over Hamiltonian terms also the soundness error of (interactive) proofs	S
r	Number of copies (repetitions) in LH verification protocol see another usage for $r(n)$ below	r
ℓ	Index over copies (repetitions) in LH verification protocol $0 \leq \ell \leq d$ indexes levels in a Merkle tree $\ell(n)$: Total length of all prover messages in an IOArg	i
m	Number of repetitions in Mahadev's protocol	k
i	Index over repetitions in Mahadev's protocol	i
$\mathcal{S}(i, \ell)$	Set of indices of the k qubits affected by Hamiltonian verification term sampled for copy i, ℓ	overloaded with Hamiltonian index S
c	Completeness; Completeness Error is $1 - c$	c
s	Soundness Error	s
$\Gamma = b - a$	Absolute promise gap for a local Hamiltonian	$b - a$
γ	Relative promise gap for a local Hamiltonian	
IOP	Interactive Oracle Proof	
IOArg	Interactive Oracle Argument	
$t(n)$	Round complexity of an IOArg	
$r(n)$	Randomness complexity of an IOArg	
$q(n)$	Query complexity of an IOArg	
$\ell(n)$	Total length of all prover messages in an IOArg	
d	Depth of a Merkle tree	
$\delta(\rho, \sigma) = \left\lVert \rho - \sigma \right\rVert_{\mathrm{tr}}$	Trace distance between density matrices ρ, σ	
$d_{\mathrm{TV}}(p, q)$	Total variation distance between distributions p and q	
$[A, B]$	Commutator of A, B i.e. $AB - BA$	
$x \Vert y$	String concatenation of strings x and y	

Table 1 Glossary of some of the mathematical notation used in this paper. When applicable, the (slightly different) notation in [ACGH20] is indicated.

A generalization of the total variation distance is the *trace distance*. To define it, let's first define the *trace norm (Schatten* 1*-norm)* of a matrix ρ as follows: $\left\lVert \rho \right\rVert_1 = \mathrm{tr}(\sqrt{\rho\rho^\dagger})$. Recall that for a density matrix ρ, it holds that $\rho = \rho^\dagger$. The trace distance between two quantum states represented by their density matrices ρ and σ is

$$\delta(\rho, \sigma) = \left\lVert \rho - \sigma \right\rVert_{\mathrm{tr}} = \frac{1}{2}\left\lVert \rho - \sigma \right\rVert_1 = \frac{1}{2}\mathrm{tr}(\sqrt{(\rho-\sigma)^2}) = \max_P \mathrm{tr}(P(\rho - \sigma)) \text{ where } P \text{ ranges over projectors.}$$

We now state some helpful propositions about the trace distance.

▶ **Proposition 18.** *The trace distance between two pure quantum states can be bounded as follows:*

$$\delta(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|) = \left\lVert |\psi\rangle\langle\psi| - |\phi\rangle\langle\phi| \right\rVert_{\mathrm{tr}} \leq \left\lVert |\psi\rangle - |\phi\rangle \right\rVert.$$

▶ **Proposition 19** (Convexity Properties of the Trace Distance; Theorem 9.3 in [NC10] and consequences thereof)**.** *Let $\{p_i\}$ and $\{q_i\}$ be probability distributions over the same index set, and $\{\rho_i\}$ and $\{\sigma_i\}$ be density operators with indices from the same index set. Then the following properties hold:*

1. ***Convexity**: $\delta(\sum_i p_i\rho_i, \sigma) \leq \sum_i p_i \delta(\rho_i, \sigma)$,*
2. ***Joint Convexity**: $\delta(\sum_i p_i\rho_i, \sum_i p_i\sigma_i) \leq \sum_i p_i \delta(\rho_i, \sigma_i)$, and*
3. ***Strong Convexity**: $\delta(\sum_i p_i\rho_i, \sum_i q_i\sigma_i) \leq \sum_i p_i \delta(\rho_i, \sigma_i) + d_{\mathrm{TV}}(p, q)$*

where $d_{\mathrm{TV}}(p,q)$ is the total variation distance between the probability distributions $\{p_i\}$ and $\{q_i\}$.

The *commutator* of two operators is given by: $[A,B] := AB - BA$. Notice that $[A,B] = -[B,A]$ and that $[A,B]^\dagger = B^\dagger A^\dagger - A^\dagger B^\dagger = [B^\dagger, A^\dagger]$. We say that two operators A, B *commute* if their commutator is 0 i.e. $[A,B] = [B,A] = 0$ and we say that they ϵ-*almost commute* if $\left\|[A,B]\right\| = \left\|[B,A]\right\| \leq \epsilon$.
If A, B are two linear operators that ϵ-almost commute, the following proposition tells us that ϵ also bounds the $\|\cdot\|$-distance between an output quantum state resulting from applying A then B on an input state and the output state had we applied B then A instead on the same input.

▶ **Proposition 20.** *If A, B are two linear operators that ϵ-almost commute, the following statements hold:*
1. *for a pure quantum state $|\psi\rangle$, it holds that (note that $\left\||\psi\rangle\right\| = 1$):*

$$\left\|AB\left|\psi\right\rangle - BA\left|\psi\right\rangle\right\| = \left\|[A,B]\left|\psi\right\rangle\right\| \leq \left\|[A,B]\right\| \cdot \left\|\left|\psi\right\rangle\right\| \leq \epsilon. \tag{1}$$

2. *for a (mixed) quantum state represented by the density matrix $\rho = \sum_i p_i \left|\psi_i\right\rangle\left\langle\psi_i\right|$, it holds that:*

$$\delta(AB\rho B^\dagger A^\dagger, BA\rho A^\dagger B^\dagger) \leq \epsilon. \tag{2}$$

Proof of Inequality (2).

$$\begin{aligned}
\delta(AB\rho B^\dagger A^\dagger, BA\rho A^\dagger B^\dagger) &= \delta\left(\sum_i p_i AB\left|\psi_i\right\rangle\left\langle\psi_i\right| B^\dagger A^\dagger, \sum_i p_i BA\left|\psi_i\right\rangle\left\langle\psi_i\right| A^\dagger B^\dagger\right) \\
&\leq \sum_i p_i \delta\left(AB\left|\psi_i\right\rangle\left\langle\psi_i\right| B^\dagger A^\dagger, BA\left|\psi_i\right\rangle\left\langle\psi_i\right| A^\dagger B^\dagger\right) && \text{by joint convexity (19)} \\
&\leq \sum_i p_i \left\|AB\left|\psi_i\right\rangle - BA\left|\psi_i\right\rangle\right\| && \text{by Proposition (18)} \\
&\leq \sum_i p_i \cdot \epsilon && \text{by Inequality (1)} \\
&= \epsilon && \text{since } \sum_i p_i = 1
\end{aligned}$$

◀

C Online Extraction of Merkle Trees in the QROM

We will now expand on Section 2.2 and show how Merkle trees can be extracted online in the quantum random oracle model relying on [DFMS22b]'s framework introduced in Section 2.2 and illustrated in Figure 3. This online extraction result is implicit in a follow-up work by [DFMS22a], but we provide a proof - with notation more relevant to our paper - which was written prior to the publication of [DFMS22a].

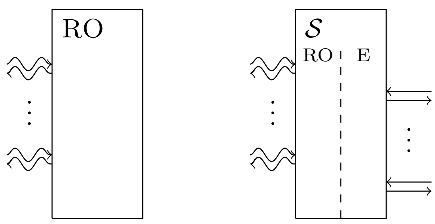


Figure 3 Figure is from [DFMS22b] and illustrates the RO interface (left) vs the extractable RO-simulator $\mathcal{S}$, with its $\mathcal{S}.RO$ and $\mathcal{S}.E$ interfaces (right). The "snaked" arrowed lines represent *quantum* queries and responses thereof, while the straight arrowed lines represent *classical* queries and responses thereof. Note that classical queries are a special case of quantum queries.

In Theorem 4.3. in [DFMS22b], multiple guarantees are proven on this simulated oracle. We cite here certain special cases of their result that we will use to prove the online extractability of Merkle trees. In [DFMS22b]'s framework, two queries are called *independent* if the input of either query does not depend on the output of the other.

▶ **Theorem 21** (Special Cases of Theorem 4.3. in [DFMS22b]). *For a $RO : \mathcal{X} \to \{0,1\}^\lambda$, the extractable RO-simulator $\mathcal{S}$ with interfaces $\mathcal{S}.RO$ and $\mathcal{S}.E$ satisfies the following properties:*

1. *If $\mathcal{S}.E$ is unused, $\mathcal{S}$ is perfectly indistinguishable from the random oracle RO.*
2. *Any two consecutive independent queries to $\mathcal{S}.RO$ commute. The same holds for $\mathcal{S}.E$.*
3. *Any two consecutive independent queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$ $8\sqrt{2^{1-\lambda}}$-almost-commute.*
4. *Classical queries to $\mathcal{S}.RO$ and $\mathcal{S}.E$ are idempotent (applying either twice in a row is equivalent to applying it once.).*
5. *The total runtime of $\mathcal{S}$ is bounded as (where q_{RO} and q_{E} are the number of queries to $\mathcal{S}.RO$ and $\mathcal{S}.E$ respectively):*

$$T_{\mathcal{S}} = O\left(q_{\mathrm{RO}} \cdot q_E + q_{\mathrm{RO}}^2\right).$$

We will also need the following proposition.

▶ **Lemma 22** (Proposition 4.5. in [DFMS22b]). *Consider a query algorithm $\mathcal{A}$ that makes q queries to $\mathcal{S}.RO$ but no query to $\mathcal{S}.E$, outputting some $t \in \{0,1\}^\lambda$ and $x \in \mathcal{X}$. Let h then be obtained by making an additional query to $\mathcal{S}.RO$ on input x. Let $\hat{x}$ be obtained by making an additional query to $\mathcal{S}.E$ on input t. Then* [12]*:*

$$\Pr_{\substack{t,x \leftarrow \mathcal{A}^{\mathcal{S}.RO} \\ h \leftarrow \mathcal{S}.RO(x) \\ \hat{x} \leftarrow \mathcal{S}.E(t)}} [\hat{x} \neq x \wedge h = t] \leq 400(q+2)^3/2^\lambda .$$

The main theorem in this section is stated in terms of a game $G_1(\lambda, d, r, q)$ illustrated in Figure 4 where a quantum adversary $\mathcal{A}$ interacts with only the RO interface of the (simulated) random oracle while a classical honest extraction algorithm $\mathcal{E}$ only (classically) interacts with the E interface of the simulated random oracle. The adversary announces a classical value rt which is *supposedly* the root of a Merkle tree and they win if they can later *"fake"* at least one of r leaves. Faking a leaf here means giving a leaf value that can be authenticated against the prior commitment, but different from that output by extraction. A referee algorithm $\mathcal{R}$ runs to determine whether the adversary won by validating the authentication paths against the root rt then comparing the adversary's leaves against the leaves given by the extraction algorithm.

The adversary - without loss of generality - can be decomposed into two quantum algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ where $\mathcal{A}_1$ makes q_1 queries to the random oracle, then announces a value rt, followed by $\mathcal{A}_2$ making q_2 queries to the random oracle, then outputs a classical string that represents their attempt to win the game where $q_1 + q_2 \leq q$. Right after $\mathcal{A}_1$ announces rt, the extraction algorithm $\mathcal{E}$ takes place and outputs $\ell = 2^d$ leaves of a Merkle tree whose root is rt. When the extraction "fails", it can default to a pre-defined leaf value (call it $\perp$) for the subtrees it failed on. The classical honest referee $\mathcal{R}$ algorithm declares that $\mathcal{A}$ won if and only if the following conditions are met:

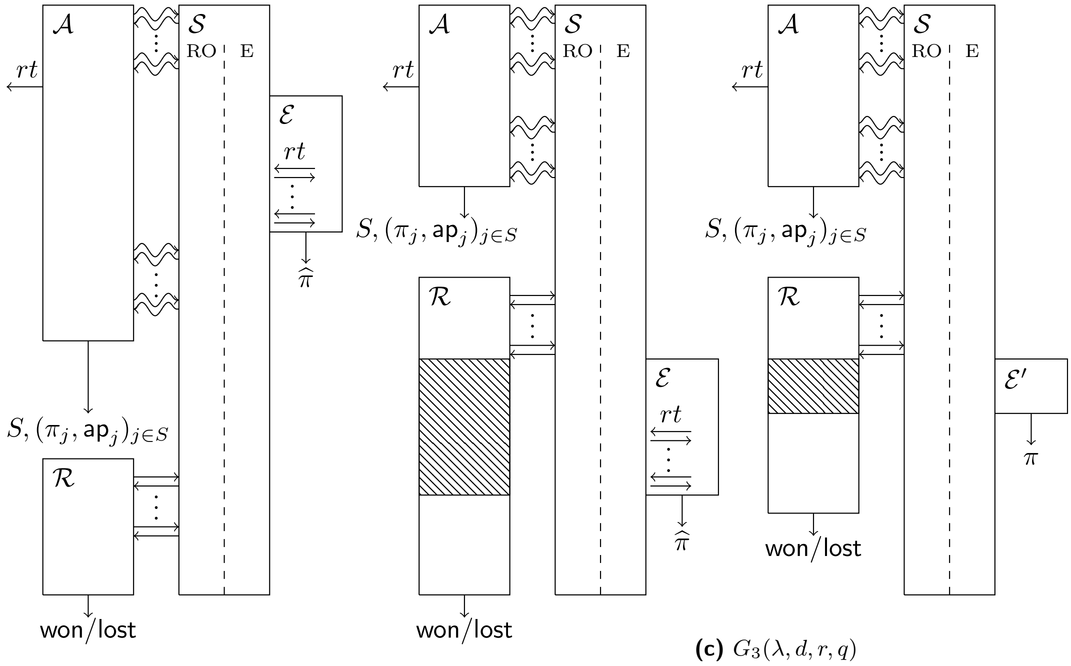
1. $\mathcal{A}_1$ outputs rt, a value in the range of the random oracle, and
2. $\mathcal{A}_2$ outputs $S, (\pi_j, \mathsf{ap}_j)_{j \in S}$ such that $S \subseteq [2^d], |S| = r$ (i.e. $\mathcal{A}$ gives r indices of the locations $\mathcal{A}$ wishes to challenge and a leaf value for each location as well as its authentication path), and $\textsc{Verify}^{\mathrm{RO}}\left(rt, S, \mathsf{ap}_{j \in S}\right)$ but $\exists j \in S : \pi_j \neq \hat{\pi}_j$ (i.e. all authentication paths are valid and consistent - see Section 2.1 - yet there is at least one location with a value different from the output of the extraction procedure).

The main theorem states that when the game G_1 is defined with the universal honest extractor and referee algorithms described earlier, any quantum adversary cannot win $G_1(\lambda, d, r, q)$ with more than a negligible probability in the security parameter λ (the number of bits in the output of the random oracle) as long as $\lambda = \omega(d)$ and the adversary makes at most $q \leq \mathrm{poly}(2^d)$ queries to the random oracle.

▶ **Theorem 4.** *For the game G_1 defined in Figure 4 by the universal referee and extractor algorithms described earlier such that $\lambda = \omega(d)$, $q \leq \mathrm{poly}(2^d)$, and any quantum adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ where $\mathcal{A}_1$ makes q_1 queries to the random oracle, then $\mathcal{A}_1$ announces a value rt, followed by $\mathcal{A}_2$ making q_2 queries to the random oracle such that $q_1 + q_2 \leq q$, then $\mathcal{A}_2$ outputs a classical string, it holds that:*

$\Pr[\mathcal{A}$ *wins* $G_1(\lambda, d, r, q)] \leq \mathrm{negl}(\lambda)$.

[12] The constant 400 is an upper bound on the constant $40e^2$ in [DFMS22b] where e is Euler's number.



(a) $G_1(\lambda, d, r, q)$ **(b)** $G_2(\lambda, d, r, q)$ **(c)** $G_3(\lambda, d, r, q)$

Figure 4 This figure illustrates the three main games used in our hybrid argument. In all of the games, $\mathcal{A}$ wins if $S \subseteq [2^d]$, $|S| = r$, and $\text{VERIFY}^{\text{RO}}\left(rt, S, \mathsf{ap}_{j\in S}\right)$, but $\exists j \in S : \pi_j \neq \widehat{\pi}_j$. The "snaked" arrowed lines represent *quantum* queries and responses thereof, while the straight arrowed lines represent *classical* queries and responses thereof. The referee $\mathcal{R}$ consists of two main procedures: (1) verifying the authentication paths which needs to interact with the $\mathcal{S}$.RO interface, and (2) comparing the output of the adversary and the extractor which does not interact with $\mathcal{S}$. The shaded rectangle indicates that the referee "pauses" its execution between these sub-procedures for the extractor execution to take place.

To prove this theorem, we give a hybrid argument outlined in Figure 4. The hybrid argument first transitions from game G_1 to game G_2 (Claim 23). The difference between games G_1 and G_2 is that the extraction procedure in G_2 happens after $\mathcal{A}_2$'s execution and the referee's oracle queries for verifying the authentication paths. Then, the argument transitions from game G_2 to game G_3 (Claim 25). The difference between games G_2 and G_3 is that in G_3 a new extractor $\mathcal{E}'$ is used which simply outputs a copy of the adversary's attempt (augmented with $\perp$ values for unchallenged leaves). Notice that no adversary can win game G_3 because of how $\mathcal{E}'$ is defined i.e. $\Pr[\mathcal{A} \text{ wins } G_3(\lambda, d, r, q)] = 0$ for any adversary! Notice that in the game G_3, it does not make a difference whether the extractor "relays" the adversary's output before or after the referee's validation of the authentication paths. Both games are equivalent in terms of the adversary's winning probability (which is 0 in either case).

We describe below how the extractor for games G_1 and G_2 works. This extraction procedure is called recursively starting with $\mathcal{E}(rt, d)$. The symbol $||$ denotes string concatenation.

$\mathcal{E}(y, d)$

- $x := \mathcal{S}.E(y)$
- If $d \stackrel{?}{=} 0$, **return** x
- Else, set $x_0 || x_1 := x$
- **return** $\mathcal{E}(x_0, d-1) || \mathcal{E}(x_1, d-1)$

On the other hand, the extractor $\mathcal{E}'$ used in game G_3 works as follows.

$\mathcal{E}'(rt, d, S, (\pi_j, \mathsf{ap}_j)_{j\in S})$

- **return** $(\hat{\pi_j})_{1\leq j\leq 2^d}$ where $\hat{\pi_j} = \pi_j$ if $j \in S$ and $\perp$ otherwise

As mentioned earlier, the first step in this hybrid argument is going from game G_1 to game G_2 which we now prove in Claim 23.

▷ Claim 23. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the final joint (adversary and random oracle) states of games G_1 and G_2 respectively. Then, the following hold:

1. $\delta(\mathcal{G}_1, \mathcal{G}_2) \leq (q + r \cdot d) \cdot 2^{d+(7-\lambda)/2}$, and consequently
2. $\Big|\Pr[\mathcal{A} \text{ wins } G_1] - \Pr[\mathcal{A} \text{ wins } G_2]\Big| \leq (q + r \cdot d) \cdot 2^{d+(7-\lambda)/2}$.

Proof. In both games, the effect of the extractor on the state of $\mathcal{S}$ (the simulated oracle) can be described by a sequence of $2^d - 1$ calls to $\mathcal{S}$.E. The adversary's behavior on its joined state with $\mathcal{S}$ can be described by a sequence of at most q quantum channels and oracle unitaries (Adv_i and $\mathcal{S}$.RO respectively) where $q_1 + q_2 \leq q$ is the total number of times the adversary calls the random oracle, split into q_1 and q_2 calls before and after announcing rt respectively. The referee's effect on the joint state of the adversary and simulated oracle is $r \cdot d$ classical queries to $\mathcal{S}$.RO. We can characterize the collective actions that the extractor, the adversary, and the referee perform on the joint state of the adversary $\mathcal{A}$ and the simulated oracle $\mathcal{S}$ in games G_1 and G_2 respectively by the following algorithms:

Net effect on the joint state of G_1	
1 :	for $i = 1, \ldots, q_1$ apply Adv_i followed by $\mathcal{S}.RO$.
2 :	$\widehat{\pi} := \mathcal{E}(rt, d)$ making $2^d - 1$ classical queries to $\mathcal{S}$.E.
3 :	for $i = q_1 + 1, \ldots, q$ apply Adv_i followed by $\mathcal{S}.RO$.
4 :	The referee applies $r \cdot d$ classical $\mathcal{S}$.RO queries .

Net effect on the joint state of G_2	
1 :	For $i = 1, \ldots, q_1$ apply Adv_i followed by $\mathcal{S}.RO$.
2 :	For $i = q_1 + 1, \ldots, q$ apply Adv_i followed by $\mathcal{S}.RO$.
3 :	The referee applies $r \cdot d$ classical $\mathcal{S}$.RO queries .
4 :	$\widehat{\pi} := \mathcal{E}(rt, d)$ making $2^d - 1$ classical queries to $\mathcal{S}$.E.

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the final joint states of the simulated oracle and adversary at the end of games G_1 and G_2 respectively. Now, we bound the distance between them using this lemma from [DFMS22b].

▶ **Lemma 24** (Special Case of Theorem 4.3. in [DFMS22b]). *Any two subsequent independent queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$ $8\sqrt{2^{1-\lambda}}$-almost-commute.*

We are commuting $2^d - 1$ classical queries to $\mathcal{S}.E$ (while preserving their order) past the execution of $\mathcal{A}_2$ involving q_2 RO-queries and the referee's $r \cdot d$ classical queries to RO. Each $\mathcal{S}$.E query made by the extractor is independent of the behavior of $\mathcal{A}_2$ and independent of the result of the referee's queries. We can use the lemma to bound the distance between $\mathcal{G}_1$ and $\mathcal{G}_2$ by successively applying the triangle inequality $(q_2 + r \cdot d) \cdot (2^d - 1)$ times to obtain:

$$\delta(\mathcal{G}_1, \mathcal{G}_2) \leq (q_2 + r \cdot d)(2^d - 1) \cdot 8\sqrt{2^{1-\lambda}} \leq 8(q + r \cdot d) \cdot 2^d \sqrt{2^{1-\lambda}} = (q + r \cdot d) \cdot 2^{d+(7-\lambda)/2}. \quad (3)$$

◀

We now show how to go from game G_2 to game G_3 in Claim 25.

▷ Claim 25.

$$\Pr[\mathcal{A} \text{ wins } G_2] \leq \Pr[\mathcal{A} \text{ wins } G_3] + 400 \cdot d \cdot r(q + 2^d + 2)^3/2^\lambda.$$

Proof. To prove this, we will go through a sequence of hybrid games G'_i where each uses the extractor $\mathcal{E}'_i$ such that $d \geq i \geq 0$. The game G_2 will be equivalent to G'_d while the game G_3 will be equivalent to G'_0. Notice how the games are indexed in descending order to make the notation easier later!

To describe the extractor $\mathcal{E}'_i$ used in these hybrid games G'_i, we will use the notation set in Section 2.1 about Merkle trees. To see the difference between G'_{i+1} and G'_i, we notice what happens in the extractor $\mathcal{E}$ from G_2. It works its way down from the root rt to all the leaves of the tree. However, the extractor of game G_3 only outputs “actual” leaves for the locations challenged by the adversary while the rest is set to $\perp$. To undergo this transition from G_2 to G_3, we work level by level from the root (top level) of the tree. For any two games G'_{i+1} and G'_i where $d > i \geq 0$:

1. The extractor $\mathcal{E}'_{i+1}$ of game G'_{i+1} will start with the values Z_{i+1} and call the extractor $\mathcal{E}(z_{j,i+1}, i+1)$ for every $z_{j,i+1}$, while
2. the extractor $\mathcal{E}'_i$ of game G'_i will do the same but starting at one level downwards. Precisely, it will start with the values Z_i and call the extractor $\mathcal{E}(z_{j,i}, i)$ for every $z_{j,i}$.

   We now give the formal description of $\mathcal{E}'_i$.

$\mathcal{E}'_i(rt, d, S, (\pi_j, ap_j)_{j\in S})$

1 : Initialize output to empty string
2 : For each $z_k \in \widehat{Z}_i$:
3 : $\quad T_k = \mathcal{E}(z_k, i)$
4 : $\quad \mathsf{output} := \mathsf{output}||T_k$
5 : **return** output

When $\mathcal{E}$ is called on $z_k = \perp$, it returns 2^i leaf values of $\perp$. Notice that in the previous codebox the merge cannot fail because each of the unique z_k is the root of its own subtree which is disjoint from the other subtrees. Furthermore, notice that while the extractor outputs the leaves at the end, it computes the intermediate nodes explicitly. This fact is going to be used in the proof of Claim 26 where we will bound the probability of winning game G'_{i+1} by that of winning G'_i as follows:

$$\Pr[\mathcal{A} \text{ wins } G'_{i+1}] \leq \Pr[\mathcal{A} \text{ wins } G'_i] + 400r(q + 2^d + 2)^3/2^\lambda.$$

Using this bound, we finalize our proof of Claim 25 by applying the triangle inequality d times from game $G_2 \equiv G'_d$ to game $G'_0 \equiv G_3$:

$$\Pr[\mathcal{A} \text{ wins } G_2] \leq \Pr[\mathcal{A} \text{ wins } G_3] + 400 \cdot d \cdot r(q + 2^d + 2)^3/2^\lambda.$$

◀

It now remains to show Claim 26.

▷ Claim 26.

$$\Pr[\mathcal{A} \text{ wins } G'_{i+1}] \leq \Pr[\mathcal{A} \text{ wins } G'_i] + 400r(q + 2^d + 2)^3/2^\lambda$$

Proof. In the extractor $\mathcal{E}'_{i+1}$, let X'_i, Z'_i be the pre-images at level i that the extractor extracts by invoking $\mathcal{S}$.E on the $(i+1)$th level and that coincide with the locations of X_i, Z_i provided by the adversary. X'_i and Z'_i will be the output of $k \leq r$ calls to $\mathcal{S}$.E on the $(i+1)$th level. The probability of winning the game G'_{i+1} can be bounded as follows:

$$\begin{aligned}
&\Pr[\mathcal{A} \text{ wins } G'_{i+1}] \\
&= \Pr[\mathcal{A} \text{ wins } G'_{i+1} \text{ and } (X_i, Z_i) = (X'_i, Z'_i)] + \Pr[\mathcal{A} \text{ wins } G'_{i+1} \text{ and } (X_i, Z_i) \neq (X'_i, Z'_i)] \\
&\leq \Pr[\mathcal{A} \text{ wins } G'_i] + \Pr[\mathcal{A} \text{ wins } G'_{i+1} \text{ and } (X_i, Z_i) \neq (X'_i, Z'_i)] \\
&\leq \Pr[\mathcal{A} \text{ wins } G'_i] + \sum_{j=1}^{k} \Pr[\mathcal{A} \text{ wins } G'_{i+1} \text{ and } (x_{j,i}, z_{j,i}) \neq (x'_{j,i}, z'_{j,i})].
\end{aligned}$$

In the last line, we applied the union bound on the events E_j where event E_j is the event that $\mathcal{A}$ wins G'_{i+1} and index j is a "mismatch". We now bound the probability $\Pr[E_j]$. Let's assume that $\mathcal{A}$ wins G'_{i+1} and $(x_{j,i}, z_{j,i}) \neq (x'_{j,i}, z'_{j,i})$ where index j is a mismatch. Since winning implies the validity and consistency of the authentication paths, we know that [13] $h(x_{j,i}, z_{j,i}) = z_{j,i+1}$ which is checked by the referee via calling $\mathcal{S}.\mathrm{RO}(x_{j,i}, z_{j,i})$. This gives rise to this event: $\mathcal{S}.RO(x_{j,i}, z_{j,i}) = z_{j,i+1}$ while $\mathcal{S}.E(z_{j,i+1}) = (x'_{j,i}, z'_{j,i})$ where $(x_{j,i}, z_{j,i}) \neq (x'_{j,i}, z'_{j,i})$. The probability of this event can be bounded by Lemma 27 below. When we invoke the Lemma 27, the query algorithm $\mathcal{Y}$ consists of the adversary and the first part of the referee i.e. $\mathcal{Y} = (\mathcal{A}, \mathcal{R}_1)$. $\mathcal{Y}$ makes at most $(q + 2^d)$ queries to RO but no queries to E. By the idempotence property of classical RO queries, we can

[13] As set in Section 2.1, we use the comma to denote a concatenation that respects left/right child order.

"artificially" insert right after the execution of $\mathcal{Y}$ another application of the RO query where the mismatch happened. We can also "move" the E query where the mismatch happened to the start of the extractor $\mathcal{E}'_{i+1}$ algorithm. This is possible at no cost because the calls of the extractor $\mathcal{E}'_{i+1}$ on the $(i+1)$th level are pairwise independent and subsequent independent E queries commute (Property 2 of Theorem 21). Finally, notice that the idempotence property of classical queries to $\mathcal{S}$.RO ensures that verifying repeated intermediate nodes is equivalent to verifying the repeated node once.

▶ **Lemma 27** (Proposition 4.5. in [DFMS22b]). *Consider a query algorithm $\mathcal{Y}$ that makes q queries to $\mathcal{S}.RO$ but no query to $\mathcal{S}.E$, outputting some $t \in \mathcal{T}$ and $x \in \mathcal{X}$. Let h then be obtained by making an additional query to $\mathcal{S}.RO$ on input x, and $\hat{x}$ by making an additional query to $\mathcal{S}.E$ on input t. Then:*

$$\Pr_{\substack{t,\, x \leftarrow \mathcal{Y}^{\mathcal{S}.RO} \\ h \leftarrow \mathcal{S}.RO(x) \\ \hat{x} \leftarrow \mathcal{S}.E(t)}} [\hat{x} \neq x \wedge h = t] \leq 400(q+2)^3/2^\lambda\,.$$

Therefore, we can conclude that:

$$\Pr[E_j] = \Pr[\mathcal{A} \text{ wins } G'_{i+1} \text{ and } (x_{j,i}, z_{j,i}) \neq (x'_{j,i}, z'_{j,i}) \text{ is a mismatch }] \leq 400(q+2^d+2)^3/2^\lambda.$$

Consequently (noting that $k \leq r$),

$$\begin{aligned} \Pr[\mathcal{A} \text{ wins } G'_{i+1}] &\leq \Pr[\mathcal{A} \text{ wins } G'_i] + \sum_{j=1}^{k} \Pr[E_j] \\ &\leq \Pr[\mathcal{A} \text{ wins } G'_i] + r \cdot 400(q+2^d+2)^3/2^\lambda. \end{aligned}$$

◀

By combining the bounds of Claim 23 and Claim 25 and using the facts that $\Pr[\mathcal{A} \text{ wins } G_3] = 0$ and $r \leq 2^d$, we obtain:

$$\begin{aligned} \Pr[\mathcal{A} \text{ wins } G_1] &\leq \Pr[\mathcal{A} \text{ wins } G_3] + (q + r \cdot d) \cdot 2^{d+(7-\lambda)/2} + 400 \cdot d \cdot r(q+2^d+2)^3/2^\lambda \\ &\leq q \cdot 2^{d+(7-\lambda)/2} + d \cdot 2^{2d+(7-\lambda)/2} + 400d(q+2^d+2)^3 \cdot 2^{d-\lambda}. \end{aligned}$$

This concludes the proof of Theorem 4 by noting that this upper bound is $\mathrm{negl}(\lambda)$ since $\lambda = \omega(d)$ and $q \leq \mathrm{poly}(2^d)$.

D Analysis of Protocol 8

D.1 Completeness of Protocol 8

▶ **Theorem 9** (Completeness of Protocol 8). *For a promise problem $A \in \mathrm{IOA}_{\mathrm{RG}_{c,s}}[t(n), \ell(n), r(n), q(n)]$ such that $c(n)$ is the completeness of the IOArg, Protocol 8 built on that IOArg also has completeness $c(n)$.*

Proof. This follows by the idempotence property of the RO interface (Property 4, Theorem 21). When the verifier $\mathcal{V}$ of Protocol 8 makes the queries to the random oracle to verify the authentication paths, they will be consistent with the classical queries that the honest prover made while generating the Merkle tree commitments. Let x be a yes instance, and $|\psi\rangle$ be the quantum state given to the honest prover $\mathcal{P}$. For brevity, let $\pi_{|Q} = \left(\pi_{1|Q_1}, \ldots, \pi_{t|Q_t}\right)$ be the locations sent by $\mathcal{P}$ and $V^{\pi_{|Q}}_{\mathrm{IOARG}}(x)$ denote the output of the IOArg verifier for the same randomness choices of $\mathcal{V}$. Then, we can compute the acceptance probability as follows:

$$\begin{aligned} \Pr[\langle \mathcal{P}, \mathcal{V}\rangle \text{ accepts } x] &= \Pr_{\pi_{|Q} \leftarrow \mathcal{P}^{|\psi\rangle}}[V^{\pi_{|Q}}_{\mathrm{IOARG}}(x) \text{ accepts and } \forall i \leq t\; \textsc{Verify}^{\mathrm{RO}}\left(rt_i, Q_i, (\mathsf{ap}_{i,j})_{j\in Q_i}\right)] \\ &= \Pr_{\pi_{|Q} \leftarrow \mathcal{P}^{|\psi\rangle}}[V^{\pi_{|Q}}_{\mathrm{IOARG}}(x) \text{ accepts }] \qquad \text{by idempotence} \\ &= \Pr[\langle \mathcal{P}^{|\psi\rangle}_{\mathrm{IOARG}}, \mathcal{V}_{\mathrm{IOARG}}\rangle \text{ accepts } x]. \end{aligned}$$

◀

D.2 Soundness of Protocol 8

▶ **Theorem 10** (Computational Soundness of Protocol 8). *Consider a promise problem A with an interactive oracle argument i.e. $A \in \mathrm{IOA}\textsc{rg}_{c,s}[t(n), \ell(n), r(n), q(n)]$. Let Protocol 8 be built on top of this IOArg in the quantum random oracle model with $\lambda = \omega(\log(\ell(n)))$. Let x be an instance of A with $n = |x|$. If a (possibly cheating) quantum prover $\mathcal{P}$ running in polynomial time $T_{\mathcal{P}}(n) = \mathrm{poly}(n)$ and access to RO can make an honest verifier $\mathcal{V}$ in such protocol accept x with probability $\geq \delta(n)$, then there exists a polynomial-time (quantum) IOArg prover $\widetilde{P}_{\mathrm{IOA}\textsc{rg}}(x)$ that can make an honest IOArg verifier accept x with probability $\geq \delta(n) - \mathrm{negl}(\lambda)$.*

Proof of Theorem 10. Consider a quantum polynomial-time prover $\mathcal{P}$ in Protocol 8 running in $T_{\mathcal{P}}(n)$ time that makes the honest verifier $\mathcal{V}$ accept on an instance x with probability $\geq \delta(n)$ where $n = |x|$. According to the protocol description, this prover $\mathcal{P}$ can be decomposed into the quantum channels $(\mathcal{P}_1, \ldots, \mathcal{P}_k, \mathcal{P}_{k+1})$ where $\mathcal{P}_i$ makes h_i queries to RO such that $\sum_{1 \leq i \leq t(n)+1} h_i \leq T_{\mathcal{P}}(n)$. Furthermore, notice that the honest verifier can be decomposed into the classical algorithms $(\mathcal{V}_1, \ldots, \mathcal{V}_{t(n)}, \mathcal{V}_{\mathcal{R}}, \mathcal{V}_{\mathrm{IOA}\textsc{rg}})$ such that:

- $\mathcal{V}_i$ is basically a relay interface connected to the incoming messages from the IOArg verifier $\widetilde{\mathcal{V}}$ (in particular $\mathcal{V}_{t(n)}$ is where the verifier sends the challenged locations),
- $\mathcal{V}_{\mathcal{R}}$ is the predicate that verifies the authentication paths of the claimed nodes, and
- $\mathcal{V}_{\mathrm{IOA}\textsc{rg}}$ is the verdict algorithm of the underlying IOArg.

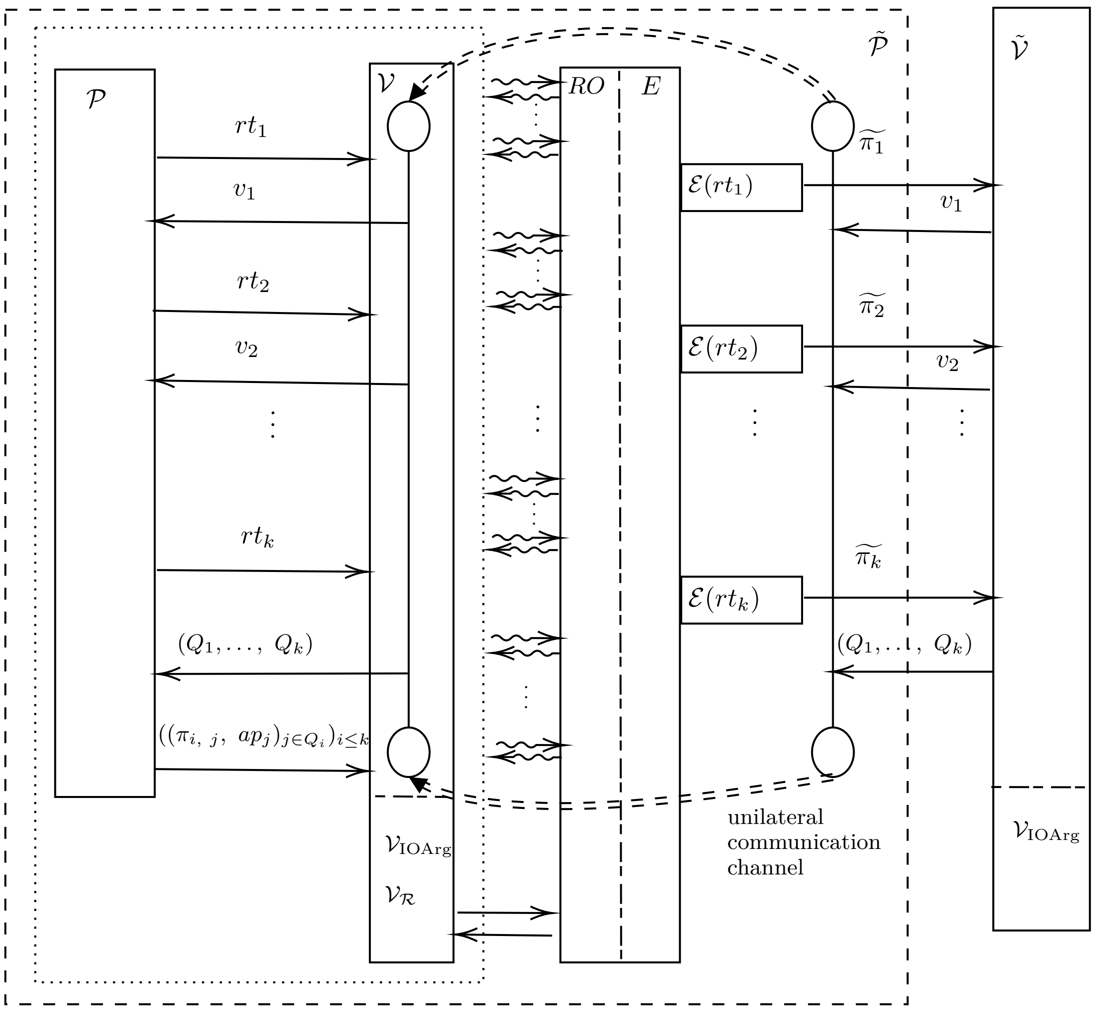


Figure 5 This figure illustrates the reduction from the succinct argument interaction $\langle \mathcal{P}, \mathcal{V} \rangle$ to a polynomial-time IOArg prover $\widetilde{\mathcal{P}}$ interacting with the honest IOArg verifier $\widetilde{\mathcal{V}}$. The prover is split into two parts: one that interacts with the E interface and one that interacts with the RO interface. Communication goes unilaterally from the former to the latter. The unilateral communication is indicated by a line with two circles at its ends. This IOArg prover can make the IOArg verifier accept the instance x with probability $\geq \delta(n) - \mathrm{negl}(\lambda)$.

As illustrated in Figure 5, we construct a (quantum) polynomial-time IOArg prover $\widetilde{\mathcal{P}}$ (in the quantum random oracle). This prover is a quantum polynomial-time interactive algorithm described by the following sequence of sub-algorithms: $\widetilde{\mathcal{P}} = \left(\widetilde{\mathcal{P}}_1, \ldots, \widetilde{\mathcal{P}}_{t(n)}\right)$. Each $\widetilde{\mathcal{P}}_i$ performs the following in order:

1. it executes $\mathcal{P}_i$ which is the corresponding action of the prover $\mathcal{P}$ in the ith round, then
2. it calls the extractor $\mathcal{E}$ with access to the $\mathcal{S}$.E interface of the simulated oracle. It will then send the extracted string $\widetilde{\pi}_i$ to the verifier $\widetilde{\mathcal{V}}$ in the form of an oracle message.

Given the description of the constructed prover $\widetilde{\mathcal{P}}$, we bound $\eta := \Pr_{\substack{rt_i \leftarrow \mathcal{P}_i(x) \\ \widetilde{\pi}_i \leftarrow \mathcal{E}(rt_i) \\ \$ \xleftarrow{\$} \{0,1\}^{r(n)}}} [\langle \mathcal{P}, \mathcal{V} \rangle \text{ accepts } x]$

$$\begin{aligned}
\eta &= \Pr[\langle \mathcal{P}, \mathcal{V} \rangle \text{ accepts } x \text{ and } \forall i\, \pi_{i|Q_i} = \widetilde{\pi}_{i|Q_i}] \\
&\quad + \Pr[\langle \mathcal{P}, \mathcal{V} \rangle \text{ accepts } x \text{ and } \exists i\, \pi_{i|Q_i} \neq \widetilde{\pi}_{|Q_i}] \qquad \text{(law of total probability)} \\
&= \Pr[\mathcal{V}_{\text{IOARG}}^{\pi_{|Q}}(x) \text{ accepts, } \mathcal{V}_{\mathcal{R}} \text{ accepts, and } \forall i\, \pi_{i|Q_i} = \widetilde{\pi}_{i|Q_i}] \\
&\quad + \Pr[\mathcal{V}_{\text{IOARG}}^{\pi_{|Q}}(x) \text{ accepts, } \mathcal{V}_{\mathcal{R}} \text{ accepts, and } \exists i\, \pi_{i|Q_i} \neq \widetilde{\pi}_{i|Q_i}] \\
&\leq \Pr[\mathcal{V}_{\text{IOARG}}^{\widetilde{\pi}_{|Q}}(x) \text{ accepts }] + \Pr[\mathcal{V}_{\mathcal{R}} \text{ accepts and } \exists i\, \pi_{i|Q_i} \neq \widetilde{\pi}_{i|Q_i}].
\end{aligned}$$

If $\Pr[\mathcal{V}_{\mathcal{R}} \text{ accepts and } \exists i\, \pi_{i|Q_i} \neq \widetilde{\pi}_{i|Q_i}] \leq \text{negl}(\lambda)$, we can conclude that:

$$\Pr[\mathcal{V}_{\text{IOARG}}^{\widetilde{\pi}_{|Q}}(x) \text{ accepts }] \geq \Pr[\langle \mathcal{P}, \mathcal{V} \rangle \text{ accepts } x] - \text{negl}(\lambda) \geq \delta(n) - \text{negl}(\lambda). \tag{4}$$

Now, it remains to show that $\Pr[\mathcal{V}_{\mathcal{R}} \text{ accepts and } \exists i\, \pi_{|Q_i} \neq \widetilde{\pi}_{|Q_i}] \leq \text{negl}(\lambda)$ which we will prove by applying Theorem 4. To do that, we notice that for each round i, we can build an adversary $\mathcal{A}^{(i)} = (\mathcal{A}_1^{(i)}, \mathcal{A}_2^{(i)})$ where $\mathcal{A}_1^{(i)} = (\mathcal{P}_1, \mathcal{V}_1, \ldots, \mathcal{P}_{i-1}, \mathcal{V}_{i-1}, \mathcal{P}_i)$ and $\mathcal{A}_2^{(i)} = \left(\mathcal{V}_i, \mathcal{P}_{i+1}, \mathcal{V}_{i+1}, \ldots, \mathcal{P}_{t(n)}\right)$ that already matches the syntax of an adversary for game $G_1\,(\lambda(n), \log(\ell_i(n)), q_i(n), h(n))$ introduced in Section C with the game parameters properly set via the parameters of the underlying IOArg (Definition 7). Indeed, we have $h(n) \leq \text{poly}(\ell_i(n))$ since $h(n) = \text{poly}(n)$ and $\ell_i(n) \leq \text{poly}(n)$. We also have $q_i(n) \leq \ell_i(n)$. Therefore, for any adversary $\mathcal{A}$ making at most $h(n)$ queries, we have:

$$\Pr[\mathcal{A} \text{ wins } G_1] \leq \text{negl}(\lambda). \tag{5}$$

Let $\mathcal{I}$ be the final state at the end of the interaction in Figure 5. Let $\mathcal{I}'$ be obtained by moving the extractors $\mathcal{E}(rt_1), \ldots, \mathcal{E}(rt_{i-1})$ past the extractor $\mathcal{E}(rt_i)$ while preserving their order. Notice that all the queries made to RO are independent of these E calls. Also, each of these extractors' chain of E-queries is independent of the queries of $\mathcal{E}(rt_i)$. Also, notice that because we are working with non-adaptive IOArgs in this paper, the behavior of $\widetilde{\mathcal{V}}$ does not depend on these calls. There are $i - 1 \leq t(n)$ extractors that we will move past at most $h(n)$ queries. Each jth extractor makes $\ell_j(n) - 1 \leq \ell(n)$ queries. Therefore, we conclude by Property 4 of Theorem 21 that:

$$\delta(\mathcal{I}, \mathcal{I}') \leq h(n) \cdot t(n) \cdot \ell(n) \cdot 8 \cdot \sqrt{2^{1-\lambda}}. \tag{6}$$

Therefore, we have:

$$\begin{aligned}
&\Pr[\mathcal{V}_{\mathcal{R}} \text{ accepts, and } \exists i\, \pi_{iQ_i} \neq \widetilde{\pi}_{iQ_i} \text{ in interaction } \mathcal{I}] \\
&\leq \Pr[\mathcal{V}_{\mathcal{R}} \text{ accepts, and } \exists i\, \pi_{iQ_i} \neq \widetilde{\pi}_{iQ_i} \text{ in interaction } \mathcal{I}'] + \delta(\mathcal{I}, \mathcal{I}') \\
&\leq \Pr[\mathcal{A} \text{ wins } G_1\,(\lambda(n), \log(\ell_i(n)), q_i(n), h(n))] + 8 \cdot t(n) \cdot h(n) \cdot \ell(n)\sqrt{2^{1-\lambda}} && \text{Inequality (6)} \\
&\leq \text{negl}(\lambda) + \text{poly}(\ell(n))\sqrt{2^{1-\lambda}} && \text{Theorem 4} \\
&\leq \text{negl}(\lambda) && \text{since } \lambda = \omega(\log(\ell(n))).
\end{aligned}$$

Finally, we need to verify that $\widetilde{\mathcal{P}}$ runs in poly(n) time as long as the underlying argument prover $\mathcal{P}$ runs in polynomial time. This is true because each of $\mathcal{P}_i$, $\mathcal{E}(rt_i)$, $\mathcal{V}_i$ run in polynomial time. Furthermore, by Property 6 of Theorem 21 the simulator $\mathcal{S}$ runs in time $T_{\mathcal{S}} = O\left(q_{\text{RO}} \cdot q_E + q_{\text{RO}}^2\right)$ where q_E and q_{RO} are the number of queries to $\mathcal{S}$.RO and $\mathcal{S}.E$ respectively. The number of queries for either type is at most poly(n) because they are made by the underlying polynomial time algorithms.

◀

D.3 Communication Complexity of Protocol 8

We analyze Protocol 8's communication complexity (excluding the setup message) provided that the underlying IOArg is parameterized as $\text{IOA}\textsc{rg}_{c,s}[t(n), \ell(n), r(n), q(n)]$. In the ith round, the prover sends a Merkle tree root which is in the range of the random oracle and therefore has length λ. The verifier sends then the message v_i which has $r_i(n)$ bits. For $t(n)$ rounds, a total of $\lambda \cdot t(n) + r(n)$ is sent so far by both the prover and verifier excluding the setup. The verifier at the end sends the $q(n)$ locations needed where each location is expressed by $\log(\ell(n))$ where $\log(\ell(n)) = O(\log(n))$ because $\ell(n) \leq \text{poly}(n)$. This means that a total of $O(q(n) \cdot \log(n))$ bits are sent by the verifier for this purpose. Finally, the prover sends the requested leaves and their authentication paths. Each authentication path is represented by $O(\log(\ell(n)) \cdot \lambda) = O(\log(n) \cdot \lambda)$ bits. Therefore, the prover sends a total of $O(q(n) \cdot \log(n) \cdot \lambda)$ bits in this round. Therefore, the total communication cost in this protocol is $O\left(\lambda \cdot (t(n) + q(n) \cdot \log(n)) + r(n)\right)$ classical bits. The resulting protocol is succinct when $q(n) = O(\text{poly}(\log(n))) = \tilde{O}(1)$, $r(n) = \tilde{O}(1)$, $t(n) = \tilde{O}(1)$, and $\ell(n) = \text{poly}(n)$.

E Modular Construction of Protocol 6

In this Appendix, we give an exposition of how to build Protocol 6 modularly. We generalize the proofs of [ACGH20] to work with any constant locality k and any promise gap function γ.

E.1 Quantum-verifier protocol for XZ local Hamiltonians

We will now give an exposition of a quantum-verifier protocol for the (n, k, γ)-LH-XZ problem which appeared in [ACGH20] and builds on earlier works of [MNS16, MF16, FHM18, VZ19]. [MF16, FHM18]'s earlier version described a proof system for QMA where the verifier is a quantum machine capable of performing X and Z measurements on a single qubit (i.e. a probabilistic classical device and a single-qubit quantum device capable of performing Pauli measurements as instructed by the classical device). The protocol starts by the verifier sampling a Hamiltonian term to be verified. The prover sends the qubits of the witness state one at a time. The verifier measures the qubits affected by the Hamiltonian term and discards the rest thus achieving this economic architecture of a single qubit. [VZ19] and [ACGH20] described parallel-repeated versions of this protocol and used them to obtain zero-knowledge argument systems for QMA. [ACGH20]'s version made another modification so that the protocol can be compiled using Mahadev's verifiable measurement protocol into a non-interactive classical-verifier version. Mahadev's protocol involves generating a pair of private/public keys that depends on the measurement basis. However, the measurement basis could depend on the Hamiltonian term since a Hamiltonian term could affect by X on a qubit while another Hamiltonian term could affect by Z on the same qubit. Therefore, they modified the protocol so that the measurement bases (X or Z) for each qubit are sampled uniformly (and therefore independent of the Hamiltonian). This way, the key generation does not depend on the Hamiltonian (but rather only on an upper bound on the number of qubits involved).

We state here [ACGH20]'s modified version but with a slight difference where we follow [MF16]'s track to only measure the qubits needed to verify the Hamiltonian while [ACGH20] measured all qubits and ignored the ones not used. Furthermore, we will parameterize the protocol for any constant k and any arbitrary relative promise gap γ.

▶ **Protocol 28** (Variant of Protocol 3 in [ACGH20]; Single-qubit verifier protocol for the local Hamiltonian problem (n, k, γ)-LH-XZ with instance-independent setup).

Parties:
1. ***Prover*** $\mathcal{P}$***:*** *A quantum polynomial-time machine that wants to convince the verifier that an input to the* (n, k, γ)-LH-XZ *problem has a groundstate of low energy i.e.* $\leq a$*.*
2. ***Verifier*** $\mathcal{V}$***:*** *A quantum polynomial-time machine that interacts with the prover to verify that an input* XZ *Hamiltonian has a groundstate of low energy.*

Parameters:
1. n*: number of qubits.*
2. r*: number of parallel repetitions of the protocol.*

Setup: $\mathcal{V}$ *samples the bases* $h_1, \ldots, h_r \leftarrow \{0,1\}^n$ *i.i.d. uniformly. Each string* h_ℓ *is an* n*-bit string where* 0 *or* 1 *mean measure the corresponding qubit in the* Z *or* X *basis respectively.*

Inputs: ***Input to both parties:*** $x = (H = \sum_{s=1}^{S} d_s H_s, a, b)$ *an instance of the* (n, k, γ)-LH-XZ *promise problem.*

Input to honest prover on yes instances: $|\Psi\rangle = |\psi\rangle^{\otimes r}$ *(r copies of $|\psi\rangle$ the ground state of the Hamiltonian H).*

Round $\mathcal{P}$: *$\mathcal{P}$ sends the witness state $|\Psi\rangle = |\psi\rangle^{\otimes r}$.*

$\mathcal{V}$'s verdict:

1. *$\mathcal{V}$ samples r i.i.d. Hamiltonian terms (one term for each copy) $s_1, \ldots, s_r \leftarrow \pi$ where the distribution π is given by:*

$$\pi(s) = \frac{|d_s|}{\sum\limits_s |d_s|}.$$

For each chosen Hamiltonian term s_ℓ, a choice of measurement bases will be imposed on at most k qubits which are acted upon by non-identity Pauli observables. Denote the set of indices of such qubits by $\mathcal{S}(\ell)$.

2. *$\mathcal{V}$ records $A \subseteq [r]$, the subset of copies where the measurements imposed by the chosen term are consistent with the random bases choices given by h. For each $\ell \in A$,*
   a. *Set $m_{\ell,j} = 1$ if $j \notin \mathcal{S}(\ell)$ i.e. the j-th qubit was acted upon by the identity in the term s_ℓ; otherwise (i.e. $j \in \mathcal{S}(\ell)$) set $m_{\ell,j}$ to the outcome of measuring it in the $h_{\ell,j}$ basis. This gives the outcomes $(m_{\ell,1}, \ldots, m_{\ell,k})$.*
   b. *$\mathcal{V}$ sets $v_\ell = \frac{1}{2}\left(1 - \textsc{sgn}(d_{s_\ell}) \cdot \prod\limits_{j \in \mathcal{S}(\ell)} m_{\ell j}\right)$ (i.e. set to 1 iff the measurement has the opposite sign of the coefficient of the selected term).*
3. *$\mathcal{V}$ accepts iff* [14] $\sum_{\ell \in A} v_\ell \geq \frac{(c+s)}{2} \cdot |A| = \frac{\left(2-(b-a)/\sum\limits_s |d_s|\right)}{4} \cdot |A|$ *where:*

$$c := \frac{1}{2} - \frac{a}{2\sum\limits_s |d_s|} \qquad \text{and} \qquad s := \frac{1}{2} - \frac{b}{2\sum\limits_s |d_s|}.$$

The following theorem establishes bounds on the completeness and soundness errors of this protocol.

▶ **Theorem 29** (Appendix B of [ACGH20]). *Let r be the number of copies used in Protocol 28 for an instance of the (n, k, γ)-LH-XZ problem, then the protocol has:*
1. *completeness error $\leq e^{-r\gamma^2/2^{k+4}}$, and*
2. *soundness error $\leq e^{-r\gamma^2/2^{k+4}}$*

where $\gamma = \frac{b-a}{S}$ is the relative promise gap as defined in Definition 5.

In Appendix E.1.1, we write down the proof of Theorem 29 which is basically a mirror of the proof of Lemma 3.1 in [ACGH20]'s Appendix B by setting the locality to k instead of 2. It suffices to take r to be any function that is $\omega(\frac{\log(n)}{\gamma^2})$ to make the completeness and soundness negligible.

▶ **Corollary 30** (Lemma 3.1. in [ACGH20]). *If $r = \omega(\frac{\log(n)}{\gamma^2})$, then Protocol 28 has negligible completeness and soundness errors.*

E.1.1 Completeness and soundness of the quantum-verifier protocol

We will now prove Theorem 29 which establishes the completeness and soundness of Protocol 28 in Section E.1. The proof is a mirror of the proof of Lemma 3.1 in Appendix B of [ACGH20] by setting the locality to k instead of 2. It also uses the proof ideas in [VZ19, MNS16].

Proof of Theorem 29. The protocol is repeated r times. For each copy, the sampled k-local Hamiltonian term will dictate that (at most) k qubits be measured in certain bases (X or Z). The randomly chosen bases for the k qubits in the protocol setup are consistent with the desired measurements with probability $\geq \frac{1}{2^k}$. Since we have r copies, there are t consistent copies with probability $\geq \binom{r}{t}(\frac{1}{2^k})^t(1 - \frac{1}{2^k})^{r-t}$.

Let X_ℓ be the binary random variable corresponding to the verdict at copy ℓ (i.e. v_ℓ). By following the computation from [MNS16], we can compute the expected value of this random variable.

[14] Notice that $c > s$ and $\frac{c+s}{2}$ is the midpoint of c and s. Therefore, another way to read this as explained in [VZ19]: $\mathcal{V}$ accepts iff $\left(\frac{1}{|A|}\sum_{\ell \in A} v_\ell\right)$ is closer to c than to s. See the appendix for the details of this computation. We suspect that there was a typo in this expression in [ACGH20].

$$\begin{aligned}
\mathbb{E}\left[X_\ell\right] &= \sum_{1\le s\le S} \frac{1}{2}\left(1 - \textsc{sgn}(d_s)\cdot \langle\psi|\, H_s\,|\psi\rangle\right)\cdot \pi(s) \\
&= \frac{1}{2}\sum_{1\le s\le S}\pi(s) - \frac{1}{2}\sum_{1\le s\le S}\pi(s)\cdot \textsc{sgn}(d_s)\cdot \langle\psi|\, H_s\,|\psi\rangle \\
&= \frac{1}{2} - \frac{1}{2}\sum_{1\le s\le S}\frac{|d_s|\textsc{sgn}(d_s)}{\sum_s |d_s|}\cdot \langle\psi|\, H_s\,|\psi\rangle \\
&= \frac{1}{2} - \frac{1}{2}\sum_{1\le s\le S}\frac{d_s}{\sum_s |d_s|}\cdot \langle\psi|\, H_s\,|\psi\rangle \\
&= \frac{1}{2} - \frac{1}{2\sum_s |d_s|}\sum_{1\le s\le S}\langle\psi|\, d_s H_s\,|\psi\rangle = \frac{1}{2} - \frac{\langle\psi|\, H\,|\psi\rangle}{2\sum_s |d_s|} \qquad (7)
\end{aligned}$$

In the "yes case" when $|\psi\rangle$ is the groundstate, we have $\mathbb{E}\left[X_\ell\right] \ge \frac{1}{2} - \frac{a}{2\sum_s |d_s|}$ because $\langle\psi|\, H\,|\psi\rangle \le a$ when $|\psi\rangle$ is the groundstate. Call this lower bound $c := \frac{1}{2} - \frac{a}{2\sum_s |d_s|}$.

In the "no case" for any state $|\psi\rangle$, we have $\mathbb{E}\left[X_\ell\right] \le \frac{1}{2} - \frac{b}{2\sum_s |d_s|}$ because $\langle\psi|\, H\,|\psi\rangle \ge b$ for any state $|\psi\rangle$. Call this upper bound $s := \frac{1}{2} - \frac{b}{2\sum_s |d_s|}$.

To bound the soundness error, let's consider the probability of acceptance in the case of a no instance. The probability that the protocol accepts conditioned on the event that the set of consistent copies was A with $|A| = t$ is given by the following:

$$\begin{aligned}
\Pr[\text{accept} \mid |A| = t] &= \Pr[\frac{1}{t}\sum_{\ell\in A} X_\ell \ge \frac{c+s}{2}] \\
&= \Pr[\frac{1}{t}\sum_{\ell\in A} X_\ell - s \ge \frac{c-s}{2}] \le e^{-tg^2/2} \qquad \text{By Hoeffding's inequality}
\end{aligned}$$

where $g = c - s$ is the absolute promise gap Γ divided by $2\sum_s |d_s|$. Now, using the fact that this event occurs with probability $\binom{r}{t}(\frac{1}{2^k})^t(1-\frac{1}{2^k})^{r-t}$, we put that together to compute the acceptance probability as follows:

$$\begin{aligned}
\Pr[\text{accept}] &= \sum_{t=0}^{r} \Pr[|A| = t]\cdot \Pr[\text{accept} \mid |A| = t] \\
&\le \sum_{t=0}^{r}\binom{r}{t}(\frac{1}{2^k})^t(1-\frac{1}{2^k})^{r-t}\cdot e^{-tg^2/2} \\
&= \sum_{t=0}^{r}\binom{r}{t}(\frac{1}{2^k}\cdot e^{-g^2/2})^t(1-\frac{1}{2^k})^{r-t} \\
&= (\frac{e^{-g^2/2}}{2^k} + 1 - \frac{1}{2^k})^r && \text{Binomial Theorem} \\
&= (\frac{e^{-g^2/2} + 2^k - 1}{2^k})^r \\
&\le (\frac{(1-g^2/4) + 2^k - 1}{2^k})^r && \text{since } e^{-x} \le 1 - x/2 \text{ for } x \in [0,1] \\
&= (\frac{-g^2/4 + 2^k}{2^k})^r = (1 - \frac{g^2}{2^{k+2}})^r \\
&\le e^{-rg^2/2^{k+2}} && \text{since } 1 - x \le e^{-x} \text{ for } x \ge 0
\end{aligned}$$

To bound the completeness error, we perform the same manipulations above to bound the probability of rejection in the case of a yes instance.

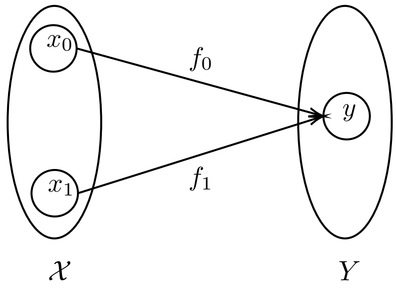


Figure 6 Claw in the functions f_0, f_1 mapping from $\mathcal{X}$ to Y

$$\Pr[\text{reject} \mid |A| = t] = \Pr[\frac{1}{t}\sum_{\ell \in A} X_\ell < \frac{c+s}{2}]$$
$$\leq \Pr[c - \frac{1}{t}\sum_{\ell \in A} X_\ell > \frac{c-s}{2}] \leq e^{-tg^2/2}$$

By performing the same manipulations, we obtain $\Pr[\text{reject}] \leq e^{-rg^2/2^{k+2}}$.

By noticing that $\sum_s |d_s| \leq S$, we can see that $g = c - s = \frac{b-a}{2\sum_s |d_s|} \geq \frac{\gamma}{2}$ where γ is the relative promise gap. We can conclude with the symmetric upper bound on the completeness and soundness errors: $e^{-r\gamma^2/2^{k+4}}$.

◀

E.2 Mahadev's verifiable measurement protocol

In 2018, Mahadev published two works [Mah18a, Mah18b] achieving the following under the computational assumption of the quantum hardness of Learning With Errors (LWE):

1. classical verification of quantum computation, and
2. classical homomorphic encryption of quantum circuits.

Part of her works' contribution was also introducing a protocol for verifiable measurement that uses a quantum-computationally binding scheme for the classical "commitment" [15] of quantum states. For a detailed description of the protocol, please refer to the original [Mah18b] paper or Section 2.2 of [VZ19] for a concise summary. Borrowing the exposition style of [VZ19, ACGH20], we are going to shed light on the verifiable measurement protocol in this subsection. A key component of the protocol is the concept of *claw-free function families*. These are function families for which it is computationally infeasible to find a *claw* except via a trapdoor. A claw as demonstrated in Figure 6 for two functions $f_0, f_1 : \mathcal{X} \to Y$ is a pair (x_0, x_1) such that $f_0(x_0) = f_1(x_1)$. Furthermore, it is computationally infeasible to find a string d and the bit $d \cdot (x_0 \oplus x_1)$ where (x_0, x_1) are part of a claw [BCM+18].

E.2.1 The Case of One Qubit

We summarize how to verifiably measure (i.e. commit and measure later) a qubit[16] $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ using the pair of functions $f_{\kappa,0}, f_{\kappa,1}$ where κ is a key sent by the verifier. Actually, the selection of the functions depends on the basis we want to perform the verifiable measurement in. This is outlined in Protocol 31 and the notation f or g will be used depending on whether we are doing Hadamard or standard basis measurement (respectively). However, in this walkthrough, we will use the letter f assuming we are interested in a Hadamard basis measurement. The prover (i.e. the measuring quantum device) performs the commitment phase by preparing the following uniform

[15] Note that this notion of binding commitment is different from the one commonly used in cryptography where the commitment needs to be hiding as well.
[16] We demonstrate how to commit to a qubit state, but the scheme can be generalized to states with more qubits.

superposition on all elements of the domain $\mathcal{X}$ and applying the function f_κ in superposition:

$$\frac{1}{\sqrt{|\mathcal{X}|}}\left(\sum_{x\in\mathcal{X}} \alpha\,|0\rangle\,|x\rangle\,|f_{\kappa,0}(x)\rangle + \sum_{x\in\mathcal{X}} \beta\,|1\rangle\,|x\rangle\,|f_{\kappa,1}(x)\rangle\right). \tag{8}$$

Expression 8 contains three quantum registers as follows:

- $|b\rangle$: the *committed qubit* register,
- $|x\rangle$: the *pre-image* register, and
- $|f_{\kappa,b}(x)\rangle$: the *commitment* or *output* register.

The prover now measures the commitment register obtaining a value $y \in \mathcal{Y}$ which is the commitment value to be sent to the verifier. This will also make the state collapse to a post-measurement state consistent with the performed measurement as follows:

$$\frac{1}{\sqrt{\#(y)}} \sum_{f_{\kappa,0}(x_0)=f_{\kappa,1}(x_1)=y} \alpha\,|0\rangle\,|x_0\rangle\,|y\rangle + \beta\,|1\rangle\,|x_1\rangle\,|y\rangle \tag{9}$$

where $\#(y)$ is the number of claws with y as their image. Notice how the original qubit state $|\psi\rangle$ (i.e. the committed qubit) is now "entangled with a superposition" of the pre-images (x_0, x_1). After the commitment phase, the verifier challenges the prover by uniformly sampling a challenge bit c and accordingly performing one of the following rounds (each w.p. 1/2):

1. test round ($c = 0$): the verifier asks the prover to measure the pre-image register and the committed qubit register in the standard basis and send back the results, or
2. Hadamard round ($c = 1$): the verifier asks the prover to measure the pre-image register and the committed qubit register in the Hadamard basis and send back the results.

After getting back the measurement results, the verifier executes the corresponding procedure as described in Protocol 31. While the test round is helpful in establishing soundness of the verifiable measurement, no measurement is learned if we undergo a test round. On the other hand, the Hadamard round helps us in learning the measurement outcome as described in Protocol 31.

▶ **Protocol 31** (Mahadev's Verifiable Measurement Suite of Algorithms). *Depending on which basis (call it h) we are interested in performing the measurement in, a function is sampled from one of the following two families of functions:*

1. ***Noisy Trapdoor Claw-free Functions (NTCFs)*** $\mathcal{F}$
   (for X (Hadamard) basis measurement; $h = 1$):

   $$\mathcal{F} = \{f_{pk} \mid f_{pk} : \{0,1\} \times \mathcal{X} \to \mathcal{D}_\mathcal{Y}\}_{pk\in\mathcal{K}_\mathcal{F}}.$$

   *This family of functions satisfy this **injective pair** property: there exists a perfect matching $\mathcal{M}_{pk} \subseteq \mathcal{X} \times \mathcal{X}$ (i.e. matching where every $x \in \mathcal{X}$ is incident to exactly one edge) such that:*

   $$(x_0, x_1) \in \mathcal{M}_{pk} \iff f_{pk}(0, x_0) = f_{pk}(1, x_1).$$

2. ***Noisy Trapdoor Injective Functions (NTIFs)*** $\mathcal{G}$
   (for Z (standard) basis measurement; $h = 0$):

   $$\mathcal{G} = \{g_{pk} \mid g_{pk} : \{0,1\} \times \mathcal{X} \to \mathcal{D}_\mathcal{Y}\}_{pk\in\mathcal{K}_\mathcal{G}}.$$

   *This family of functions satisfy this **injectivity** property:*

   $$(x, b) \neq (x', b') \Rightarrow \mathsf{supp}\ g_{pk}(b, x) \cap \mathsf{supp}\ g_{pk}(b', x') = \emptyset.$$

The following algorithms are used in Mahadev's protocol:

- ***Trapdoor Inversion:*** $(z, e) = \mathsf{Inv}_\mathcal{F}(f_{pk}, sk, b, y)$ *[similarly defined for $(z, e) = \mathsf{Inv}_\mathcal{G}(g_{pk}, sk, b, y)$]. This is a deterministic algorithm that can assign to* $\mathbf{e}$ *a pre-image such that $y \in \mathsf{supp}(f_{pk}(b, e))$ if this pre-image exists. In that case, z is set to 1; otherwise, it assigns 0 to z.*
- ***TestRound:*** $z = \mathsf{TestCheck}(pk, b, x, y)$ *outputs 1 iff (b, x) is a pre-image of y under the mapping f_{pk} (or g_{pk}).*
- ***HadRound:*** $(e, z) = \mathsf{HadRound}(sk, b, x, y, h)$ *takes as input a secret key sk and the measured registers b, x, y as well as a basis choice h. Depending on the basis choice, the verifier executes one of these to output the measurement:*

1. *if* $h = 0$ *(i.e. Z basis measurement is requested), output* $(e, z) \leftarrow \mathsf{Inv}_{\mathcal{G}}(g_{pk}, sk, b, y)$*.*
2. *if* $h = 1$ *(i.e. X basis measurement is requested), compute both* pre-images $x_{0,y}, x_{1,y}$*:*
   - $(z_0, x_{0,y}) = \mathsf{Inv}_{\mathcal{F}}(f_{pk}, sk, 0, y)$
   - $(z_1, x_{1,y}) = \mathsf{Inv}_{\mathcal{F}}(f_{pk}, sk, 1, y)$

   and set $e = x \cdot (x_{0,y} \oplus x_{1,y}) \oplus b$*.* z *is set to* 0 *if any of the two runs reject or if* x *is trivial (e.g.* $= 0$*); otherwise* z *is set to* 1*.*

The following theorem summarizes the soundness property of the Mahadev protocol.

▶ **Theorem 32** (Soundness of Mahadev's verifiable measurement protocol; Claim 7.1. in [Mah18b] following the exposition of Claim 2.12. in [VZ19])**.** *Under the LWE assumption, let* $\widetilde{\mathcal{P}}$ *be any (possibly cheating) quantum polynomial-time prover interacting with an honest verifier of Protocol 31 with the basis choice* h *adopting the following notation for brevity:*

- $1 - p_{h,H}$*: the probability that the verifier accepts the prover* $\widetilde{\mathcal{P}}$ *in a Hadamard round of the protocol with basis* h*,*
- $1 - p_{h,T}$*: the probability that the verifier accepts the prover* $\widetilde{\mathcal{P}}$ *in a test round of the protocol with basis* h*, and*
- $D_{\widetilde{\mathcal{P}},h}$*: the distribution over measurement outcomes obtained by the honest verifier on executing a Hadamard round with the prover* $\widetilde{\mathcal{P}}$ *for basis* h*.*

Then, there exists a negligible function μ*, a quantum state* ξ*, and a prover* $\widehat{\mathcal{P}}$ *with the following distributions:*

1. $D_{\widehat{\mathcal{P}},h}$*: the distribution over measurement outcomes obtained by an honest verifier on executing a Hadamard round with the prover* $\widehat{\mathcal{P}}$*, and*
2. $D_{\xi,h}$*: the distribution over measurement outcomes obtained by directly performing a quantum* h*-basis measurement on the state* ξ*.*

such that:

$$d_{TV}\left(D_{\widetilde{\mathcal{P}},h}, D_{\widehat{\mathcal{P}},h}\right) \leq \sqrt{p_{h,T}} + p_{h,H} + \mu \qquad \text{and} \qquad D_{\widehat{\mathcal{P}},h} \approx_c D_{\xi,h}$$

where $\approx_c$ *denotes quantum-computational indistinguishability.*

E.3 Classical-verifier argument for XZ local Hamiltonians

This was provided as Protocol 6 in Section 2.4.

▶ **Theorem 33** (Section 4 of [ACGH20])**.** *Under the LWE assumption and for a given set of parameters* $\lambda \geq n, r, m$*, and a constant* k*, Protocol 6 for the* (n, k, γ)-LH-XZ *problem has:*

1. *completeness error* $\leq \mu + \mathrm{negl}(\lambda)$*, and*
2. *soundness error* $\leq 2^{-m} + (\mu)^{1/4} + \mathrm{negl}(\lambda)$

where $\mu = e^{-r\gamma^2/2^{k+4}}$ *is the symmetric bound on the completeness and soundness errors of Protocol 28 in Theorem 29.*

▶ **Corollary 34** (Theorem 4.6. in [ACGH20])**.** *Under the LWE assumption, for every constant* k*, Protocol 6 with* $\lambda \geq n$*,* $r = \omega(\frac{\log(n)}{\gamma^2})$ *and* $m = \omega(\log(n))$ *has negligible completeness and soundness errors.*

F Soundness of ACGH's protocol after eliminating redundancy

We now analyze the soundness of Protocol 12 given in Section 8. Most of the contents that follow in this Appendix except Lemma 40 and its proof are verbatim or almost verbatim from [ACGH20] while changing whatever is needed and proving Lemma 40 that we give.

▶ **Theorem 35** (Mirror of Section 4 of [ACGH20])**.** *Under the LWE assumption, for a given set of parameters* $\lambda \geq n, r, m$*, and a constant* k*, Protocol 12 for the* (n, k, γ)-LH-XZ *problem has:*

1. *completeness error* $\leq \mu + \mathrm{negl}(\lambda)$*, and*
2. *soundness error* $\leq 2^{-m} + (\mu)^{1/4} + \mathrm{negl}(\lambda)$

where $\mu \leq e^{-r\gamma^2/2^{k+4}}$ *is the symmetric bound on the completeness and soundness errors of Protocol 28 in Theorem 29.*

▶ **Lemma 36** (Mirror of Lemma 4.4. in [ACGH20]). *In Protocol 12 parameterized by positive integers r and m, let $\{U_c\}_{c\in\{0,1\}^m}$ be any set of unitaries that may be implemented by $\mathcal{P}$ after the challenge coins are sent. Let $|\Psi_{pk}\rangle$ be any state that $\mathcal{P}$ holds in the commitment round, and suppose $\mathcal{P}$ applies U_c followed by honest measurements when the coins are c. Then there exists a negligible function δ such that $\mathcal{V}_1,\ldots,\mathcal{V}_m$ accept $\mathcal{P}$ with probability at most $2^{-m}+\mu^{1/4}+\delta^{1/2}$ where $\mu = e^{-r\gamma^2/2^{k+4}}$ is the soundness error of Protocol 28 with r copies.*

Proof. The success probability of any prover in the k-fold protocol is

$$\Pr[\text{success}] = 2^{-m} \mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h),h,s}[\langle\Psi_{pk}|\sum_c \pi^{U_c}_{s,sk,c}|\Psi_{pk}\rangle]$$

where h, s are drawn from uniform distributions. The uniform string s is used in [ACGH20] to sample the Hamiltonian terms from the distribution induced by the coefficients of the terms.

▶ **Lemma 37.** *(Lemma 4.3. verbatim from [ACGH20]). Let $A_1,\ldots,A_m$ be projectors and $|\psi\rangle$ be a quantum state. Suppose there are real numbers $\delta_{ij}\in[0,2]$ such that $\langle\psi|A_iA_j+A_jA_i|\psi\rangle\le\delta_{ij}$ for all $i\neq j$. Then $\langle\psi|A_1+\cdots+A_m|\psi\rangle\le 1+\left(\sum_{i<j}\delta_{ij}\right)^{1/2}$.*

Exactly as in [ACGH20], define a total ordering on $\{0,1\}^m$ such that $a<b$ if $a_i<b_i$ for the smallest index i such that $a_i\neq b_i$. Then by Lemma 37, we have

$$\Pr[\text{success}] \le 2^{-m} + 2^{-m}\mathop{\mathbb{E}}_{h,s}\left[\sum_{a<b}\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\pi^{U_a}_{s,sk,a}\pi^{U_b}_{s,sk,b}+\pi^{U_b}_{s,sk,b}\pi^{U_a}_{s,sk,a}|\Psi_{pk}\rangle]\right]^{1/2}.$$

▶ **Lemma 38** (Modified Lemma 4.2. in [ACGH20]). *Let $\mathcal{P}$ be a prover in Protocol 12 that prepares $|\Psi_{pk}\rangle$ in Round $\mathcal{P}_1$ and performs U_c in Round $\mathcal{P}_2$. Let $a,b\in\{0,1\}^m$ such that $a\neq b$ and choose i such that $a_i\neq b_i$. Then there is an (mr)-qubit quantum state ρ such that for every basis choice h and randomness s,*

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\pi^{U_b}_{s,sk,b}\pi^{U_a}_{s,sk,a}+\pi^{U_a}_{s,sk,a}\pi^{U_b}_{s,sk,b}|\Psi_{pk}\rangle\right]\le 2\alpha^{1/2}_{h_i,s_i,\rho}+\mathrm{negl}(n)\,,$$

where $\alpha_{h_i,s_i,\rho}$ is the success probability with ρ conditioned on the event that h_i is sampled.

By Lemma 38, there exists a negligible function δ such that

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\pi^{U_a}_{s,sk,a}\pi^{U_b}_{s,sk,b}+\pi^{U_b}_{s,sk,b}\pi^{U_a}_{s,sk,a}|\Psi_{pk}\rangle]\le 2\alpha^{1/2}_{h_{i(a,b)},\rho_{ab}}+\delta(n)$$

for every pair (a,b). Here $i(a,b)$ is the smallest index i such that $a_i\neq b_i$ and ρ_{ab} is the reduced quantum state associated with a,b, as guaranteed by Lemma 38.

Let μ be the soundness error of the Protocol 28 with r copies. We have

$$\begin{aligned}\Pr[\text{success}] &\le 2^{-m}+2^{-m}\mathop{\mathbb{E}}_{h,s}\left[\sum_{a<b}\left(2\alpha^{1/2}_{h_{i(a,b)},s_{i(a,b)},\rho_{ab}}+\delta(n)\right)\right]^{1/2}\\ &\le 2^{-m}+\mu^{1/4}+\sqrt{\delta(n)} \qquad \text{see [ACGH20] for the computations.}\end{aligned}$$

◀

To prove Lemma 38, we will again follow [ACGH20]'s proof and replace the projectors Π with the new projectors π and using this modified version of Lemma 4.1. in [ACGH20].

▶ **Lemma 39** (Modified Version of Lemma 4.1. in [ACGH20]). *Let $\mathcal{P}=(|\Psi_{pk}\rangle, U_{\mathfrak{t}}, U_{\mathfrak{h}})$ be a prover in Protocol 12 such that, for every $h\in\{0,1\}^{nr}$ and $s\in\{0,1\}^p$ (p is a polynomial bound on the bits needed to sample the Hamiltonian terms),*

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}}|\Psi_{pk}\rangle]\ge 1-\mathrm{negl}(n)\,. \tag{10}$$

Then there exists an (nr)-qubit quantum state ρ such that, for every h, s,

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}}|\Psi_{pk}\rangle]\le\alpha_{h,s,\rho}+\mathrm{negl}(n)\,,$$

where $\alpha_{h,s,\rho}$ is the success probability in Protocol 28 with basis choice h and r-copies of the quantum state ρ.

Proof of lemma 39. We use the following helpful technical lemma that we show later:

▶ **Lemma 40.** *Let $\pi_1, \dots, \pi_n$ be single qubit projectors on the same domain. Let P_1 and P_2 be of the form $\bigotimes_{i=1}^n \widehat{\pi}_i$ where $\widehat{\pi}_i$ is either I or π_i. **If** for some $|\phi\rangle$, it holds that:*

$$\langle\phi| P_1 |\phi\rangle \geq 1 - \delta_1 \text{ and } \langle\phi| P_2 |\phi\rangle \geq 1 - \delta_2$$

then, it follows that:

$$\langle\phi| P_2 P_1 |\phi\rangle \geq 1 - (\delta_1 + \delta_2).$$

Noting that $\Pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}}$ in Lemma 4.1. in [ACGH20] is the same as $\prod_s \pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}}$, and since each $\pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}}$ is of the form in the hypothesis of Lemma 40, we can apply Lemma 40 for as many as there are Hamiltonian terms and obtain:

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow \mathsf{Gen}(1^\lambda,h)} [\langle\Psi_{pk}| \Pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}} |\Psi_{pk}\rangle] \geq 1 - O(n) \cdot \mathrm{negl}(n)$$

Now this is basically the hypothesis of Lemma 4.1 in [ACGH20]. Therefore, the first paragraph of the proof of this Lemma holds but the second paragraph is the one that is slightly different. In this alteration, we consider the new measurement $\{\pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}}, \mathbb{1} - \pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}}\}$. Verbatim from [ACGH20], the proof completes by noting that these two cases are computationally indistinguishable:

1. An output is sampled from the distribution $D_{\mathcal{P},h}$ and the verifier applies the final checks in Protocol 28. In this case, the final outcome is obtained by performing the measurement $\{\pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}}, \mathbb{1} - \pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}}\}$ on the state $|\Psi_{pk}\rangle$, and accepting if the first outcome is observed.
2. An output is sampled from the distribution $D_{\rho,h}$ and the verifier applies the final checks in Protocol 28. In this case, the acceptance probability is $\alpha_{h,s,\rho}$ by the protocol definition.

We can conclude that:

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow \mathsf{Gen}(1^\lambda,h)} [\langle\Psi_{pk}| \pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}} |\Psi_{pk}\rangle] \leq \alpha_{h,s,\rho} + \mathrm{negl}(n)\,,$$

◀

We now move to prove lemma 40.

Proof of lemma 40. Let $\{|u_0\rangle, |u_1\rangle\}$ be an orthonormal basis for the domain of each projector π_i. $|\phi\rangle$ can be written as:

$$|\phi\rangle = \sum_{b\in\{0,1\}^n} \alpha_b |u_{b_1} \dots u_{b_n}\rangle \text{ where } \sum_{b\in\{0,1\}^n} |\alpha_b|^2 = 1$$

We write $P_j = \bigotimes_{i=1}^n \widehat{\pi}_{j,i}$. We use $v_{i,b}$ to denote $\langle u_b| \pi_i |u_b\rangle$ and $\widehat{v}_{j,i,b}$ to denote $\langle u_b| \widehat{\pi}_{j,i} |u_b\rangle$. It can be seen that:

$$\widehat{v}_{j,b} := \langle u_b| \bigotimes_{i=1}^n \widehat{\pi}_{j,i} |u_b\rangle = \widehat{v}_{j,1,b} \dots \widehat{v}_{j,n,b}.$$

By the hypothesis of the lemma, we have:

$$\begin{aligned}
\langle\phi| P_j |\phi\rangle &= \sum_{b\in\{0,1\}^n} |\alpha_b|^2 \langle u_b| \bigotimes_{i=1}^n \widehat{\pi}_{j,i} |u_b\rangle \\
&= \sum_{b\in\{0,1\}^n} |\alpha_b|^2 \langle u_{b_1}| \widehat{\pi}_{j,1} |u_{b_1}\rangle \dots \langle u_{b_n}| \widehat{\pi}_{j,n} |u_{b_n}\rangle \\
&= \sum_{b\in\{0,1\}^n} |\alpha_b|^2 \,\widehat{v}_{j,1,b} \dots \widehat{v}_{j,n,b} \\
&= \sum_{b\in\{0,1\}^n} |\alpha_b|^2 \,\widehat{v}_{j,b} \geq 1 - \delta_j
\end{aligned}$$

Let's write $\widehat{\Pi}_i = \widehat{\pi}_{2,i}\widehat{\pi}_{1,i}$. Since π_i is a projector, so is $\pi_i^2 = \pi_i$. Therefore, $\widehat{\Pi}_i$ is either π_i or I. Let $\widehat{v}_{i,b} := \langle u_b| \widehat{\Pi}_i |u_b\rangle$ and for brevity let $\widehat{v}_b := \langle u_b| \bigotimes_{i=1}^n \widehat{\Pi} |u_b\rangle$. By the fact that $\langle u_b| \pi_i |u_b\rangle \leq$

$\langle u_b | I | u_b \rangle = 1$, one can conclude, by exhausting all cases, that $\widehat{v}_{i,b} \geq \widehat{v}_{2,i,b}\widehat{v}_{1,i,b}$ and consequently $\widehat{v}_b \geq \widehat{v}_{2,b}\widehat{v}_{1,b}$. Putting this together, it follows that:

$$\begin{aligned}
\langle \phi | P_2 P_1 | \phi \rangle &= \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \langle u_b | \bigotimes_{i=1}^{n} \widehat{\Pi}_i | u_b \rangle \\
&= \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \langle u_{b_1} | \widehat{\Pi}_1 | u_{b_1} \rangle \dots \langle u_{b_n} | \widehat{\Pi}_n | u_{b_n} \rangle \\
&= \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_b
\end{aligned}$$

Now, let's show that $\langle \phi | P_2 P_1 | \phi \rangle \geq 1 - (\delta_1 + \delta_2)$ which is equivalent to $1 - \langle \phi | P_2 P_1 | \phi \rangle \leq \delta_1 + \delta_2$.

$$\begin{aligned}
1 - \langle \phi | P_2 P_1 | \phi \rangle &= 1 - \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_b \\
&\leq 1 - \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{2,b}\widehat{v}_{1,b} && (\widehat{v}_b \geq \widehat{v}_{2,b}\widehat{v}_{1,b}) \\
&\leq \left(\delta_1 + \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{1,b} \right) - \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{2,b}\widehat{v}_{1,b} && (\langle \phi | P_1 | \phi \rangle \geq 1 - \delta_1) \\
&= \delta_1 + \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{1,b} \, (1 - \widehat{v}_{2,b}) \\
&\leq \delta_1 + \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, (1 - \widehat{v}_{2,b}) && (\widehat{v}_{1,b} \leq 1) \\
&= \delta_1 + \sum_{b \in \{0,1\}^n} |\alpha_b|^2 - \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{2,b} \\
&= \delta_1 + \left(1 - \sum_{b \in \{0,1\}^n} |\alpha_b|^2 \, \widehat{v}_{2,b} \right) \\
&\leq \delta_1 + \delta_2 && (\langle \phi | P_2 | \phi \rangle \geq 1 - \delta_2)
\end{aligned}$$

◀

G A Tale of Alice on a Quantum Island

Alice is a passionate explorer who studied Egyptology and cryptology. She has just embarked on an expedition to the island of Elephantine. Legend has it that the ancient Egyptians built a large-scale quantum computer on this very island 4,000 years ago. While she was excavating for this elusive quantum computer, she found a hieroglyphic LATEXpapyrus entitled *"proof of the quantum PCP theorem and reductions to XZ Hamiltonians"*! "What a fruitful trip already!", Alice said to herself as she continued her excavation. After a few days, she found herself in front of a wondrous building and a sign carved in Hieroglyphics that says "The Classical Interface". "Is this a bottle of liquid luck [17] or water?", Alice exclaimed looking at her water bottle after realizing that she just unveiled an ancient instantiation of a *quantum random oracle*! Alice goes around the giant building to find another hieroglyphic sign on the other side that says "The Quantum Interface". Suddenly, someone appears in a blue cloak while facing towards the entrance and waving aggressively with his hand in front of the building as if he were casting a sequence of spells. As Alice calls on him, he turns and she immediately recognizes him as Merlin! After a short conversation, Merlin claims to have access to the ancient Egyptian quantum computer! While it seems like good news, he also claims that he magically hid it with no intention of unveiling it to anyone. However, not all hope is lost because he claims to be able to communicate with it using his magical powers. Alice has a lot of important questions about life, the universe, and everything that she hopes to settle with the help of this long-awaited quantum computer. She even designed an efficient quantum circuit to answer these quests in anticipation of this very moment. Although Merlin promises to help her,

[17] Also called Felix Felicis for the interested reader; c.f. J.K. Rowling (2005).

she is concerned that he might mislead her. As a well-trained cryptographer, Alice asks Merlin to prove to her that indeed these answers were obtained by executing her quantum circuit. She asks him to engage with her in an interactive conversation where she will ask him follow-up questions. Merlin agrees to Alice's proposal on one condition; "If you do not trust me, that is your problem. I am very thirsty at the moment. I will only respond to these follow-up questions if and only if my answers to these additional questions are very short." Merlin said to Alice unhappily. Alice knew that she could reasonably suggest to him to drink as much as he desires from the Nile flowing right in front of them. However, she did not feel that she had the luxury to further upset him. Since Alice is a very smart cryptographer who read this paper, she knows how to verify Merlin's answers to her questions under some assumptions despite his short temper!

References

AALV09 Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The detectability lemma and quantum gap amplification. *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09*, 2009.

ACGH20 Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 153–180. Springer, 2020.

BCM+18 Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, Oct 2018.

BCS16 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 31–60, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

BDF+11 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

BKL+22 James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. Succinct classical verification of quantum computation. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2022.

BL08 Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A*, 78(1), Jul 2008.

BM22 James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 15:1–15:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

BR93 Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, page 62–73, New York, NY, USA, 1993. Association for Computing Machinery.

CCY20 Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 181–206. Springer, 2020.

Chi05 Andrew M. Childs. Secure assisted quantum computation. *Quantum Inf. Comput.*, 5(6):456–466, 2005.

CM21 Lijie Chen and Ramis Movassagh. Quantum merkle trees, 2021.

CM22 Lijie Chen and Ramis Movassagh. Making quantum local verifiers simulable with potential applications to zero-knowledge. *arXiv preprint arXiv:2209.10798*, 2022.

CMS19 Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, pages 1–29, 2019.

CMSZ21 Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 49–58. IEEE, 2021.

CVZ20 Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for qma, with preprocessing. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 799–828. Springer, 2020.

DFMS22a Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Efficient nizks and signatures from commit-and-open protocols in the QROM. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 729–757. Springer, 2022.

DFMS22b Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 677–706. Springer, 2022.

FHM18 Joseph Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Physical Review Letters*, 120(4), jan 2018.

GJMZ22 Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. Commitments to quantum states. Cryptology ePrint Archive, Paper 2022/1358, 2022. `https://eprint.iacr.org/2022/1358`.

GKK18 Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Theory of Computing Systems*, 63(4):715–808, jul 2018.

Gri18 Alex Bredariol Grilo. *Quantum proofs, the local Hamiltonian problem and applications. (Preuves quantiques, le problème des Hamiltoniens locaux et applications)*. PhD thesis, Sorbonne Paris Cité, France, 2018.

Kil92 Joe Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 723–732, New York, NY, USA, 1992. Association for Computing Machinery.

Mah18a Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 332–338. IEEE Computer Society, 2018.

Mah18b Urmila Mahadev. Classical Verification of Quantum Computations. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 259–267, 2018.

MF16 Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

MNS16 Tomoyuki Morimae, Daniel Nagaj, and Norbert Schuch. Quantum proofs can be verified using only single-qubit measurements. *Physical Review A*, 93(2), Feb 2016.

NC10 M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

TMT22 Yuki Takeuchi, Tomoyuki Morimae, and Seiichiro Tani. Sumcheck-based delegation of quantum computing to rational server. *Theoretical Computer Science*, 924:46–67, 2022.

Unr16a Dominique Unruh. Collapse-binding quantum commitments without random oracles. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 166–195, 2016.

Unr16b Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

VZ19 Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *IACR Cryptology ePrint Archive*, 2019:194, 2019.

Zha19 Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO*

2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019.

Zha21 Jiayu Zhang. Succinct blind quantum computation using a random oracle. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1370–1383. ACM, 2021.

Zha22 Jiayu Zhang. Classical verification of quantum computations in linear time. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 46–57. IEEE, 2022.