Succinct Classical Verification of Quantum Computation

James Bartusek*, Yael Tauman Kalai**, Alex Lombardi***, Fermi Ma†, Giulio Malavolta‡, Vinod Vaikuntanathan§, Thomas Vidick¶, and Lisa Yang‖

Abstract. We construct a classically verifiable *succinct* interactive argument for quantum computation (BQP) with communication complexity and verifier runtime that are poly-logarithmic in the runtime of the BQP computation (and polynomial in the security parameter). Our protocol is secure assuming the post-quantum security of indistinguishability obfuscation (iO) and Learning with Errors (LWE). This is the first succinct argument for quantum computation *in the plain model*; prior work (Chia-Chung-Yamakawa, TCC '20) requires both a long common reference string and non-black-box use of a hash function modeled as a random oracle.

At a technical level, we revisit the framework for constructing classically verifiable quantum computation (Mahadev, FOCS '18). We give a self-contained, modular proof of security for Mahadev's protocol, which we believe is of independent interest. Our proof readily generalizes to a setting in which the verifier's first message (which consists of many public keys) is *compressed*. Next, we formalize this notion of compressed public keys; we view the object as a generalization of constrained/programmable PRFs and instantiate it based on indistinguishability obfuscation.

Finally, we compile the above protocol into a fully succinct argument using a (sufficiently composable) succinct argument of knowledge for NP.

Using our framework, we achieve several additional results, including

- Succinct arguments for QMA (given multiple copies of the witness),
- Succinct *non-interactive* arguments for BQP (or QMA) in the quantum random oracle model, and
- Succinct batch arguments for BQP (or QMA) assuming post-quantum LWE (without iO).

1 Introduction

Efficient verification of computation is one of the most fundamental and intriguing concepts in computer science, and lies at the heart of the P vs. NP question.

* UC Berkeley. Email: bartusek.james@gmail.com.
** Microsoft Research and MIT. Email: yael@microsoft.com.
*** MIT. Email: alexjl@mit.edu.
† Simons Institute and UC Berkeley. Email: fermima@alum.mit.edu.
‡ MPI-SP. Email: giulio.malavolta@hotmail.it.
§ MIT. Email: vinodv@mit.edu.
¶ Caltech. Email: vidick@caltech.edu.
‖ MIT. Email; lisayang@mit.edu.

It has been studied in the classical setting for over three decades, giving rise to beautiful notions such as interactive proofs [GMR85], multi-prover interactive proofs [BGKW88], probabilistically checkable proofs [BFL90,ALM+92,AS92], and culminating with the notion of a *succinct* (interactive and non-interactive) *argument* [Kil92,Mic94]. Roughly speaking, a succinct argument for a T-time computation enables a prover running in $\mathsf{poly}(T)$ time to convince a $\mathsf{polylog}(T)$-time verifier of the correctness of the computation using only $\mathsf{polylog}(T)$ bits of communication, with soundness against all polynomial-time cheating provers.

In a breakthrough result in 2018, Mahadev [Mah18] presented an interactive argument system that enables a classical verifier to check the correctness of an arbitrary *quantum* computation. Mahadev's protocol represents a different kind of interactive argument — unlike the traditional setting in which the prover simply has more *computational resources* (i.e., running time) than the verifier, the prover in Mahadev's protocol works in a qualitatively more powerful *computational model*. More precisely, for any T-time quantum computation, Mahadev's protocol enables a quantum prover running in time $\mathsf{poly}(T)$ to convince a classical $\mathsf{poly}(T)$-time verifier with $\mathsf{poly}(T)$ bits of classical communication. Soundness holds against all quantum polynomial-time cheating provers under the post-quantum hardness of the learning with errors (LWE) problem.

A fundamental question is whether we can get the best of both worlds: can the prover have *both* a more powerful computational model *and* significantly greater computational resources? Namely, we want an interactive argument system for T-time quantum computation in which the quantum prover runs in $\mathsf{poly}(T)$ time and convinces a $\mathsf{polylog}(T)$-time classical verifier with $\mathsf{polylog}(T)$ bits of classical communication.

We answer this question affirmatively, both for $\mathsf{poly}(T)$-time quantum computations, corresponding to the complexity class **BQP**, and also for the non-deterministic analog **QMA**.

Theorem 1.1 (Succinct Arguments for BQP). *Let λ be a security parameter. Assuming the existence of a post-quantum secure indistinguishability obfuscation scheme* (iO) *and the post-quantum hardness of the learning with errors problem* (LWE)*, there is an interactive argument system for any T-time quantum computation on input x,*[1] *where*

- *the prover is quantum and runs in time* $\mathsf{poly}(T, \lambda)$*,*
- *the verifier is classical and runs in time* $\mathsf{poly}(\log T, \lambda) + \tilde{O}(|x|)$*,*[2] *and*
- *the protocol uses* $\mathsf{poly}(\log T, \lambda)$ *bits of classical communication.*

Theorem 1.2 (Succinct Arguments for QMA). *Assuming the existence of a post-quantum secure indistinguishability obfuscation scheme* (iO) *and the*

[1] A T-time quantum computation is a *language* L decidable by a bounded-error T-time quantum Turing machine [BV97]. We leave it to future work to address more complex tasks such as *sampling* problems (as in [CLLW20]).

[2] As in the classical setting, some dependence on $|x|$ is necessary at least to read the input; as in [Kil92], we achieve a fairly minimal $|x|$-dependence.

post-quantum hardness of the learning with errors problem (LWE)*, there is an interactive argument system for any T-time quantum computation on input x and a* $\mathsf{poly}(T)$*-qubit witness, where*

- *the prover is quantum and runs in time* $\mathsf{poly}(T, \lambda)$*, using polynomially many copies of the witness,*[3]
- *the verifier is classical and runs in time* $\mathsf{poly}(\log T, \lambda) + \tilde{O}(x)$*, and*
- *the protocol uses* $\mathsf{poly}(\log T, \lambda)$ *bits of classical communication.*

A New Proof of Security for the [Mah18] Protocol. One might hope to prove Theorems 1.1 and 1.2 by treating the Mahadev result as a "black box" and showing that *any* (classical) interactive argument for quantum computations can be compressed into a succinct protocol via a suitable cryptographic compiler. This is especially appealing given the extremely technical nature of Mahadev's security proof. Unfortunately, for reasons that will become clear in the technical overview, this kind of generic compilation seems unlikely to be achievable in our setting. Even worse, there does not appear to be any easily formalized property of the Mahadev protocol that would enable such a compilation.

Instead, our solution consists of two steps.

(1) We build a modified variant of the [Mah18] protocol and give an entirely self-contained proof of security. This modified protocol satisfies a few technical conditions that the original [Mah18] does not; most prominently, the *first verifier message* of our modified protocol is already succinct.
(2) We give a generic compiler that converts the protocol from Step (1) into a succinct argument system.

Our Step (1) also results in a self-contained proof of security of the original [Mah18] protocol that is more modular and amenable to further modification and generalization, which we believe will be useful for future work. Our analysis builds upon [Mah18] itself as well as an alternative approach described in Vidick's (unpublished) lecture notes [Vid20]. A concrete consequence of our new proof is that one of the two "hardcore bit" security requirements of the main building block primitive ("extended noisy trapdoor claw-free functions") in [Mah18] is not necessary.

Additional Results. Beyond our main result of succinct arguments for **BQP** and **QMA**, we explore a number of extensions and obtain various new protocols with additional properties.

- *Non-Interactive*: Although our protocols are not public-coin, we show how to modify them in order to apply the Fiat-Shamir transformation and round-collapse our protocols. As a result, we obtain designated-verifier non-interactive arguments for **BQP** (and the non-deterministic analog **QMA**) with security in the quantum random oracle model (QROM).

[3] We inherit the need for polynomially-many copies of the witness from prior works. This is a feature common to all previous classical verification protocols, and even to the quantum verification protocol of [FHM18].

- *Zero-Knowledge*: We show how to lift both variants of our protocol (interactive and non-interactive) to achieve zero-knowledge. We show a generic transformation based on classical two-party computation for reactive functionalities that makes our protocols simulatable. This transformation does not add any new computational assumption to the starting protocol.
- *Batch Arguments from LWE*: For the case of batch arguments, i.e., where the parties engage in the parallel verification of n statements, we show a succinct protocol that only assumes the post-quantum hardness of LWE (without iO). In this context, succinctness requires that the verifier's complexity scales with the size of a *single instance*, but is independent of n.

Prior Work. As discussed above, Mahadev [Mah18] constructs a *non-succinct* argument system for **BQP**/**QMA** under LWE. The only prior work addressing *succinct* classical arguments for quantum computation is the recent work of Chia, Chung and Yamakawa [CCY20]. [CCY20] constructs a classically verifiable argument system for quantum computation in the following setting:

- The prover and verifier share a $\mathsf{poly}(T)$-bits long, structured reference string (which requires a trusted setup to instantiate) along with a hash function h (e.g. SHA-3).
- The "online communication" of the protocol is succinct ($\mathsf{poly}(\log T)$).
- Security is heuristic: it can be proved when h is modeled as a random oracle, but the *protocol description itself* explicitly requires the code of h (i.e. uses h in a non-black-box way).

We specifically note that when viewed in the *plain model* (i.e., without setup), the verifier must send the structured reference string to the prover, resulting in a protocol that is *not succinct*. We note that [CCY20] was specifically optimizing for a *two-message* protocol, but their approach seems incapable of achieving succinctness in the plain model even if further interaction is allowed.

By contrast, our succinct interactive arguments are in the plain model and are secure based on well-formed cryptographic assumptions, and our succinct 2-message arguments are proved secure in the QROM (and do not require a long common reference string).

Finally, we remark that our approach to achieving succinct arguments fundamentally (and likely necessarily) differs from [CCY20] because we manipulate the "inner workings" of the [Mah18] protocol; by contrast [CCY20] makes "black-box" use of a specific soundness property of the [Mah18] protocol (referred to as "computational orthogonality" by [ACGH20]) and is otherwise agnostic to how the protocol is constructed.

Acknowledgments. AL is supported in part by a Charles M. Vest fellowship. GM is partially supported by the German Federal Ministry of Education and Research BMBF (grant 16K15K042, project 6GEM). TV is supported by AFOSR YIP award number FA9550-16-1-0495, a grant from the Simons Foundation (828076, TV), MURI Grant FA9550-18-1-0161, the NSF QLCI program through

grant number OMA- 2016245 and the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028). AL, VV, and LY are supported in part by DARPA under Agreement No. HR00112020023, a grant from the MIT-IBM Watson AI, a grant from Analog Devices and a Microsoft Trustworthy AI grant. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA. LY was supported in part by an NSF graduate research fellowship.

2 Technical Overview

Our starting point is Mahadev's protocol for classical verification of quantum computation [Mah18], the core ingredient of which is a *measurement protocol*.

2.1 Recap: Mahadev's Measurement Protocol

We begin by reviewing Mahadev's N-qubit measurement protocol. In Mahadev's protocol, a quantum prover holding an N-qubit quantum state $\boldsymbol{\rho}$ interacts with a classical verifier, who wants to obtain the result of measuring $\boldsymbol{\rho}$ according to measurement bases $h \in \{0,1\}^N$ (h_i specifies a basis choice for the ith qubit, with $h_i = 1$ corresponding to the Hadamard basis and $h_i = 0$ corresponding to the standard basis).

Trapdoor Claw-Free Functions. At the heart of the protocol is a cryptographic primitive known as an *injective/claw-free trapdoor function* (a variant of lossy trapdoor functions [PW08,PVW08,GVW15]), which consists of two trapdoor function families Inj (for injective) and Cf (for claw-free), with the following syntactic requirements:[4]

- Each function in $\mathsf{Cf} \cup \mathsf{Inj}$ is indexed by a public-key pk, where functions $f_{\mathsf{pk}} \in \mathsf{Inj}$ are injective and functions $f_{\mathsf{pk}} \in \mathsf{Cf}$ are two-to-one. Moreover, pk can be sampled along with a secret key sk that enables computing f_{pk}^{-1} (i.e., $f_{\mathsf{pk}}^{-1}(y)$ consists of a single pre-image if $f_{\mathsf{pk}} \in \mathsf{Inj}$, and two pre-images if $f_{\mathsf{pk}} \in \mathsf{Cf}$).
- All functions in Inj and Cf have domain $\{0,1\}^{\ell+1}$ (for some ℓ) and the two pre-images of y under $f_{\mathsf{pk}} \in \mathsf{Cf}$ are of the form $(0, x_0)$ and $(1, x_1)$ for some $x_0, x_1 \in \{0,1\}^{\ell}$.

An injective/claw-free trapdoor function must satisfy the following security properties:[5]

[4] The actual syntactic requirements are somewhat more complex due to the fact that the functions in question are probabilistic.

[5] In fact, Mahadev's proof relies on two different hardcore bit properties, but we show in this work that only the adaptive hardcore bit property is needed.

1. **Claw-Free/Injective Indistinguishability.** A random function in $f_{\mathsf{pk}} \leftarrow \mathsf{Cf}$ is computationally indistinguishable from a random function $f_{\mathsf{pk}} \leftarrow \mathsf{Inj}$.
2. **Adaptive Hardcore Bit.** Given $f_{\mathsf{pk}} \leftarrow \mathsf{Cf}$, it is computationally infeasible to output both (1) a pair (x, y) satisfying $f_{\mathsf{pk}}(x) = y$ and (2) a non-zero string $d \in \{0,1\}^{\ell+1}$ such that $d \cdot (1, x_0 \oplus x_1) = 0$, where $(0, x_0)$ and $(1, x_1)$ are the two preimages of y.[6]

To build some intuition about the usefulness of such function families, notice that they can be used to commit to a single classical bit quite easily. The commitment key is a function $f_{\mathsf{pk}} \in \mathsf{Inj}$, and commitment to a bit b is $y = f_{\mathsf{pk}}(b, x)$ for a random $x \in \{0,1\}^n$. It is not hard to verify that this is a statistically binding and computationally hiding commitment of b. On the other hand, if $f_{\mathsf{pk}} \in \mathsf{Cf}$, it is a statistically hiding and computationally binding commitment of b.[7]

Protocol Description. With this intuition in mind, we now describe a (slightly simplified version of) the Mahadev measurement protocol. Mahadev's protocol is a many-fold sequential repetition of the following one-bit challenge protocol.

V → P: The verifier samples N pairs of functions and their trapdoors: for every $i \in [\ell]$, he samples $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Inj}$ if $h_i = 0$ and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Cf}$ if $h_i = 1$. He sends the N-tuple of public keys (function descriptions) $\underline{(\mathsf{pk}_1, \ldots, \mathsf{pk}_N)}$ to the prover.

P → V: The prover holds an N-qubit state $\boldsymbol{\rho}$ on register $\mathcal{B} = \mathcal{B}_1 \otimes \cdots \mathcal{B}_N$. She initializes registers $\mathcal{X} = \mathcal{X}_1 \otimes \cdots \mathcal{X}_N$ so that each $\mathcal{X}_i$ is an ℓ-qubit register in uniform superposition. Then she evaluates each f_{pk_i} on the $\ell + 1$ qubit register $\mathcal{B}_i \otimes \mathcal{X}_i$ and measures the output to obtain y_i. Finally, she sends $\underline{(y_1, \ldots, y_N)}$ to the verifier, keeping the registers $\mathcal{B}, \mathcal{X}$ for herself.

V → P: The verifier picks a random challenge bit $\underline{c \leftarrow \{0,1\}}$ and sends it to the prover. $c = 0$ indicates a "test round" and $c = 1$ indicates a "measurement round".

P → V: If $c = 0$ ("test"), the prover measures each $\mathcal{B}_i \otimes \mathcal{X}_i$ in the standard basis, obtaining $(b_i, x_i) \in \{0,1\} \times \{0,1\}^\ell$, and sends $\underline{(b_i, x_i)_{i=1}^N}$ to the verifier.
If $c = 1$ ("measure"), she measures each $\mathcal{B}_i \otimes \mathcal{X}_i$ in the Hadamard basis (i.e., she applies the Hadamard gate H on each qubit and then measures in the standard basis), obtaining $d_i \in \{0,1\}^{\ell+1}$, and sends $\underline{(d_i)_{i=1}^N}$ to the verifier.

V (output): If $c = 0$ ("test"), **V** either accepts or rejects the interaction but outputs nothing. In particular, **V** checks that $f_{\mathsf{pk}_i}(b_i, x_i) = y_i$. If any of the checks fail, reject.
If $c = 1$ ("measure"), proceed as follows. Let $R \subseteq [N]$ be the set of indices where $h_i = 1$, corresponding to Hadamard basis measurements, and let $S = [N] \setminus R$ be the set of indices corresponding to standard basis measurements. For each $i \in R$, **V** computes the two inverses $(0, x_{i,0})$ and $(1, x_{i,1})$ of y_i (which are guaranteed to exist since $f_{\mathsf{pk},i} \in \mathsf{Cf}$) using sk_i. **V** sets $u_i :=$

[6] The full definition places a slightly stronger restriction on d than simply being non-zero. However, this simplified version will suffice for this overview.

[7] In particular, $f_{\mathsf{pk}} \in \mathsf{Cf}$ satisfies Unruh's definition of *collapse-binding* [Unr16].

$d \cdot (1, x_{i,0} \oplus x_{i,1})$ as the i^{th} measurement outcome. For every $i \in S$, $\mathbf{V}$ ignores d_i, and sets v_i to be the first bit of $f^{-1}_{\mathsf{pk}_i}(y_i)$, computed using the trapdoor sk_i (this is well-defined since $f_{\mathsf{pk},i} \in \mathsf{Inj}$). Finally $\mathbf{V}$ outputs the N-bit string $(u, v) \in \{0,1\}^R \times \{0,1\}^S$.

Mahadev [Mah18] proves that if a malicious prover $\mathbf{P}^*$ passes the test round with probability 1, then there exists an N-qubit quantum state $\boldsymbol{\rho}^*$ — *independent* of the verifier's measurement basis h — such that the result of measuring $\boldsymbol{\rho}^*$ according to h is computationally indistinguishable from the verifier's N-bit output distribution in the measurement round.[8] While her definition requires that such a $\boldsymbol{\rho}^*$ *exists*, Vidick and Zhang [VZ21] showed that Mahadev's proof steps implicitly define an extractor that efficiently produces $\boldsymbol{\rho}^*$ using black-box access to $\mathbf{P}^*$.

2.2 Defining a (Succinct) Measurement Protocol

Our first (straightforward but helpful) step is to give an explicit definition of a *commit-and-measure protocol* that abstracts the completeness and soundness properties of Mahadev's measurement protocol as established in [Mah18,VZ21]. Roughly speaking, a commit-and-measure protocol is sound if, for any malicious prover $\mathbf{P}^*$ that passes the test round with probability 1 and any basis choice h, there exists an efficient extractor that (without knowledge of h) interacts with prover and outputs an extracted state $\boldsymbol{\tau}$ such that the following are indistinguishable:

- the distribution of verifier outputs obtained in the measurement round from interacting with $\mathbf{P}^*$ using basis choice h, and
- the distribution of measurement outcomes obtained from measuring $\boldsymbol{\tau}$ according to h.

This abstraction will be particularly helpful for reasoning about our eventual *succinct* measurement protocols, which will necessitate modifying Mahadev's original protocol.

Can a Measurement Protocol be Succinct? Given the definition of a measurement protocol, an immediate concern arises with respect to obtaining succinct arguments: the verifier's *input* to the measurement protocol – the basis vector h – is inherently non-succinct. Since the number of qubits N grows with the runtime of the BQP computation when used to obtain quantum verification [FHM18], this poses an immediate problem.

Our solution to this problem is to only consider basis vectors h that are *succinct*; our formalization is that h must be the truth table of an efficiently

[8] This can be extended to provers that pass the test round with probability $1 - \varepsilon$ by the gentle measurement lemma. In particular, an efficient distinguisher can only distinguish the verifier's output distribution from the result of measuring some $\boldsymbol{\rho}^*$ with advantage $\mathsf{poly}(\varepsilon)$.

computable function $f : [\log N] \to \{0,1\}$. For any such h, we can represent the verifier's input as a circuit C that computes h, removing the above obstacle.

However, in order for there to be any hope of this idea working, it must be the case that measurement protocols for bases with succinct representations are still useful for constructing delegation for BQP. Fortunately, it has been shown [ACGH20] that classically verifiable (non-succinct) arguments for BQP can be constructed by invoking Mahadev's measurement protocol (and, by inspection of the proof, any measurement protocol satisfying our definition) on a *uniformly random basis string* $h \leftarrow \{0,1\}^N$. Then, by computational indistinguishability, it is also possible to use a *pseudorandom* string h that has a succinct representation, i.e., $h = (\mathsf{PRF}_s(1), \ldots, \mathsf{PRF}_s(N))$ for some (post-quantum) pseudorandom function PRF.

Thus, we focus for the moment on constructing a succinct measurement protocol for h with succinct representation, and return to the full delegation problem later.

2.3 Constructing a Verifier-Succinct Measurement Protocol

Inspecting the description of the [Mah18] protocol, there are three distinct reasons that the protocol is not succinct:

1. The verifier's first message, which consists of N TCF public keys, is non-succinct.
2. The prover's two messages, consisting of the commitments y_i and openings z_i respectively, are non-succinct.
3. The verifier's decision predicate, as it is a function of these commitments and openings, requires $\mathsf{poly}(N)$ time to evaluate.

The latter two issues turn out to be not too difficult to resolve (although there is an important subtlety that we discuss later); for now, we focus on resolving (1), which is our main technical contribution. Concretely, we want to construct a measurement protocol for succinct bases h where the verifier's first message is succinct.

Idea: Compress the Verifier's message with iO. Given the problem formulation, a natural idea presents itself: instead of having V send over N i.i.d. public keys pk_i, perhaps V can send a succinct program PK that contains the description of N public keys pk_i that are in some sense "pseudoindependent!" Using the machinery of obfuscation and the "punctured programs" technique [SW14], it is straightforward to write down a candidate program for this task: simply obfuscate the following code.

Here, C is an efficient circuit with truth table h, and $\mathsf{Gen}(1^\lambda, \mathsf{mode})$ indicates sampling either from Inj or Cf depending on whether $h_i = C(i) = 0$ or $h_i = C(i) = 1$.

Letting PK denote an obfuscation of the above program, V could send PK to P and allow the prover to compute each $\mathsf{pk}_i = \mathsf{PK}(i)$ on its own, and the protocol could essentially proceed as before, except that the verifier will have to expand its PRF seed s into $(\mathsf{sk}_1, \ldots, \mathsf{sk}_N)$ in order to compute its final output.

Input: index $i \leq N$
Hardwired Values: Puncturable PRF seed s. Circuit C.

- Compute $\mathsf{mode} = C(i)$ and $r = \mathsf{PRF}_s(i)$.
- Compute $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(1^\lambda, \mathsf{mode}; r)$.
- Output pk_i.

Problem: Proving Soundness. While it is not hard to describe this plausible modification to the [Mah18] protocol that compresses the verifier's message, it is very unclear how to argue that the modified protocol is sound. The obfuscation literature has no shortage of proof techniques developed over the last 10 years, but since we have made a "non-black-box" modification of the [Mah18] protocol, a deep understanding of the [Mah18] proof of soundness is required in order to understand to what extent these techniques are compatible with the application at hand.

We believe it *should* be possible to incorporate punctured programming techniques into Mahadev's proof of soundness in [Mah18] and conclude the desired soundness property of the new protocol. However, doing so would result in an extremely complex proof that would require the reader to verify the entirety of the [Mah18] (already very complicated) original security proof with our modifications in mind.

2.4 Proof of Soundness

Given the complicated nature of the [Mah18] proof of soundness, we instead give a *simpler* and *more modular* proof of soundness for the [Mah18] measurement protocol. Moreover, we give this proof for a generic variant of the [Mah18] protocol where the prover is given an arbitrary representation PK of N TCF public keys and show that precisely two properties of this representation PK are required in order for the proof to go through:

- An appropriate generalization of the "dual-mode" property of individual TCFs must hold for PK: for any two circuits C_1, C_2, it should be that PK_1 generated from basis C_1 is computationally indistinguishable from PK_2 generated from basis C_2. In fact, a stronger variant of this indistinguishability must hold: it should be the case that $\mathsf{PK}_1 \approx_c \mathsf{PK}_2$ even if the distinguisher is given all secret keys sk_j such that $C_1(j) = C_2(j)$.
- For every i, the adaptive hardcore bit property of f_{pk_i} should hold *even given* sk_j for all $j \neq i$.

Since these two properties are (essentially) all that is required for our proof to go through, in order to obtain a verifier-succinct protocol, it suffices to show that

the obfuscated program PK above satisfies these two properties, which follows from standard techniques.

Thus, we proceed by describing our new soundness proof for the [Mah18] measurement protocol, which transparently generalizes to the verifier-succinct setting.

The "Operational Qubits" Approach. Let P^* denote a prover that passes the test round (i.e., makes the verifier accept on the 0 challenge) with probability 1. Our goal is to show that the prover in some sense "has an N-qubit state" such that measuring this state in the h-bases produces the same (or an indistinguishable) distribution as the verifier's protocol output, which we will denote $D_{P^*,\mathrm{Out}}$. This N-qubit state should be efficiently computable from the prover's internal state $|\psi\rangle$; specifically, we use $|\psi\rangle$ to denote the prover's state after its first message y has been sent.

In order to show this, taking inspiration from [Vid20],[9] we will proceed in two steps:

1. Identify N "operational qubits" within $|\psi\rangle$. That is, we will identify a set of $2N$ observables $Z_1, \ldots, Z_N, X_1, \ldots, X_N$ (analogous to the "Pauli observables" $\sigma_{z,1}, \ldots, \sigma_{z,N}, \sigma_{x,1}, \ldots \sigma_{x,N}$) such that measuring $|\psi\rangle$ with these observables gives the outcome distribution $D_{P^*,\mathrm{Out}}$.

   Provided that these $2N$ observables roughly "behave like" Pauli observables with respect to $|\psi\rangle$ (e.g. satisfy the X/Z uncertainty principle), one could then hope to:
2. Extract a related state $|\psi'\rangle$ such that measuring $|\psi'\rangle$ in the *actual* standard/Hadamard bases matches the "pseudo-Pauli" $\{Z_j\}$, $\{X_i\}$, measurements of $|\psi\rangle$ (and therefore $D_{P^*,\mathrm{Out}}$).

Relating the Verifier's Output to Measuring $|\psi\rangle$. Our current goal is to achieve Step (1) above. Let $|\psi\rangle$ denote P^*'s post-commitment state and let U denote the unitary such that P^*'s opening is a measurement of $U|\psi\rangle$ in the Hadamard basis.

Now, let us consider the verifier's output distribution. The ith bit of the verifier's output when $h_i = 1$ is defined to be $d \cdot (x_{0,i} \oplus x_{1,i})$ (where d is the opening sent by the prover) of $U|\psi\rangle$ in the Hadamard basis. For each such i, we can define an observable X_i characterizing this measurement, that *roughly* takes the form

$$X_i \approx U^\dagger (H_{\mathcal{Z}_i} \otimes \mathsf{Id}) \left(\sum_d (-1)^{d \cdot (1, x_{0,i} \oplus x_{1,i})} |d\rangle\langle d|_{\mathcal{Z}_i} \otimes \mathsf{Id}_{\mathcal{I}, \{\mathcal{Z}_j\}_{j \neq i}} \right) (H_{\mathcal{Z}_i} \otimes \mathsf{Id}) U.$$

[9] [Vid20] gives a soundness proof for a variant of the [Mah18] protocol, but in a qualitatively weaker setting. [Vid20] only proves indistinguishability of N-qubit measurements that are either *all* in the standard basis or *all* in the Hadamard basis, and only proves indistinguishability with respect to *linear* tests of the distribution (that is, [Vid20] proves small-bias rather than full indistinguishability). Both of these relaxations are unacceptable in our setting, and achieving the latter specifically requires a different proof strategy.

Here we have slightly simplified the expression for X_i for the sake of presentation; the correct definition of X_i (see the full version) must account for the case where d is rejected by the verifier. To reiterate, the observable X_i is a syntactic interpretation of the verifier's output m_i as a function of $|\psi\rangle$.

On the other hand, when $h_i = 0$, the verifier's output m_i is not *a priori* a measurement of $|\psi\rangle$; indeed, the verifier ignores the prover's second message and just inverts y_i. However, under the assumption that the prover P^* passes the test round with probability $1 - \mathsf{negl}(\lambda)$, making use of the fact that f_{pk_i} is injective, this y_i-inverse must be equal to what the prover *would* have sent in the test round. This defines another observable on $|\psi\rangle$ that we call Z_i:

$$Z_i = \sum_{b,x} (-1)^b \, |b, x\rangle\langle b, x|_{\mathcal{Z}_i} \otimes \mathsf{Id}_{\mathcal{I}, \{\mathcal{Z}_j\}_{j \neq i}}.$$

Finally, note that the operator Z_i syntactically makes sense even when $h_i = 1$. However, X_i cannot even be *defined* when f_{pk_i} is injective, corresponding to $h_i = 0$, since X_i explicitly requires *two* inverses of y_i. Therefore, from now on, we sample all $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Cf}$ (forcing all TCFs to be 2-to-1).

This brings us to the punchline of this step: by invoking a computational assumption (the indistinguishability of Cf and Inj), we can define observables (X_i, Z_i) for all $i \in [N]$ such that for *every* i and *every* basis choice h, the distribution resulting from measuring $|\psi\rangle$ with X_i (resp. Z_i) matches the ith bit of the verifier's output distribution.

With a little more work, one can actually show that the verifier's *entire* output distribution in the h-basis is computationally indistinguishable from the following distribution $D_{P^*,\text{2-to-1}}$:

- Sample keys $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Cf}$. Run P^* to obtain $y, |\psi\rangle$.
- For each i such that $h_i = 0$, measure the first bit of the prover's ith response register in the standard basis to obtain (and output) a bit b_i.
- Measure $U|\psi\rangle$ in the Hadamard basis, obtaining strings $(d_1, \ldots, d_N)$.
- For each i such that $h_i = 1$, compute (and output) $d_i \cdot (1, x_{0,i} \oplus x_{1,i})$.

Aside: Why are these Z_j and X_i helpful? As alluded to earlier, this approach is inspired by *operational* definitions of "having an N-qubit state," which consists of a state $|\psi\rangle$ and $2N$ "pseudo-Pauli" observables $Z_1, \ldots, Z_N, X_1, \ldots X_N$ that behave "like Pauli observables" on $|\psi\rangle$. For example, it is possible to prove that many of the "Pauli group relations" hold *approximately* on these X_i, Z_j with respect to $|\psi\rangle$, meaning that (for example)

$$\langle\psi| \, Z_i X_i Z_i + X_i \, |\psi\rangle = \mathsf{negl}(\lambda)$$

and

$$\langle\psi| \, Z_j X_i Z_j - X_i \, |\psi\rangle = \mathsf{negl}(\lambda)$$

for $i \neq j$. In fact, these relations turn out to *encode* the two basic properties of the TCF f_{pk_i}: the adaptive hardcore bit property (encoded in the first relation) and

that f_{pk_i} is indistinguishable from injective[10] (encoded in the second relation)! We will not directly prove the relations here, but they are implicit in our full security proof and are the motivation for this proof strategy.

The Extracted State. Given these protocol observables $Z_1, \ldots, Z_N, X_1, \ldots, X_N$, it remains to implement Step (2) of our overall proof strategy: extracting a state $|\psi'\rangle$ whose standard/Hadamard measurement outcomes match $D_{P^*,\mathrm{Out}}$. At a high level, this is achieved by "teleporting" the state $|\psi\rangle$ onto a fresh N-qubit register in a way that *transforms* the "pseudo-Paulis" $\{X_i\}, \{Z_j\}$ into *real* Pauli observables $\{\sigma_{x,i}\}, \{\sigma_{z,j}\}$.

Fix a choice of $\{X_i, Z_i\}, |\psi\rangle \leftarrow \mathsf{Samp}$. For ease of notation, write $\mathcal{H} = \mathcal{Z} \otimes \mathcal{I} \otimes \mathcal{U}$ so that $|\psi\rangle \in \mathcal{H}$. We would like an efficient extraction procedure that takes as input $|\psi\rangle \in \mathcal{H}$ and generates an N-qubit state $\boldsymbol{\tau}$ such that, roughly speaking, measuring $|\psi\rangle$ with X/Z and measuring $\boldsymbol{\tau}$ with σ_X/σ_Z produce indistinguishable outcomes.

Intuition for the Extractor. Before we describe our extractor, we first provide some underlying intuition. For an arbitrary N-qubit Hilbert space, let $\sigma_{x,i}/\sigma_{z,i}$ denote the Pauli σ_x/σ_z observable acting on the ith qubit. For each $r, s \in \{0,1\}^N$, define the N-qubit Pauli "parity" observables

$$\sigma_x(r) \coloneqq \prod_{i:r_i=1} \sigma_{x,i} \ , \ \sigma_z(s) \coloneqq \prod_{i:r_i=1} \sigma_{z,i}.$$

Suppose for a moment that $|\psi\rangle \in \mathcal{H}$ is *already* an N-qubit state (i.e., $\mathcal{H}$ is an N-qubit Hilbert space) and moreover, that each X_i/Z_i observable is simply the corresponding Pauli observable $\sigma_{x,i}/\sigma_{z,i}$. While these assumptions technically trivialize the task (the state already has the form we want from the extracted state), it will be instructive to **write down an extractor that "teleports" this state into another N-qubit external register.**

We can do this by initializing two N-qubit registers $\mathcal{A}_1 \otimes \mathcal{A}_2$ to $|\phi^+\rangle^{\otimes N}$ where $|\phi^+\rangle$ is the EPR state $(|00\rangle + |11\rangle)/\sqrt{2}$ (the ith EPR pair lives on the ith qubit of $\mathcal{A}_1$ and $\mathcal{A}_2$). Now consider the following steps, which are inspired by the (N-qubit) quantum teleportation protocol

1. Initialize a $2N$-qubit ancilla $\mathcal{W}$ to $\left|0^{2N}\right\rangle$, and apply $H^{\otimes 2N}$ to obtain the uniform superposition.
2. Apply a "controlled-Pauli" unitary, which does the following for all $r, s \in \{0,1\}^N$ and all $|\phi\rangle \in \mathcal{H} \otimes \mathcal{A}_1$:
$$|r,s\rangle_{\mathcal{W}} |\phi\rangle_{\mathcal{H},\mathcal{A}_1} \to |r,s\rangle_{\mathcal{W}} (\sigma_x(r)\sigma_z(s)_{\mathcal{H}} \otimes \sigma_x(r)\sigma_z(s)_{\mathcal{A}_1}) |\phi\rangle_{\mathcal{H},\mathcal{A}_1}$$
3. Apply the unitary that XORs onto $\mathcal{W}$ the outcome of performing N Bell-basis measurements[11] on $\mathcal{A}_1 \otimes \mathcal{A}_2$ onto $\mathcal{W}$, i.e., for all $u, v, r, s \in \{0,1\}^N$:
$$|u,v\rangle_{\mathcal{W}} (\sigma_x(r)\sigma_z(s) \otimes \mathsf{Id})_{\mathcal{A}_1,\mathcal{A}_2} \left|\phi^+\right\rangle^{\otimes N}_{\mathcal{A}_1,\mathcal{A}_2}$$

[10] Technically, the property encoded is the *collapsing* of f_{pk_i}, which is implied by (but not equivalent to) being indistinguishable from injective.

[11] The Bell basis consists of the 4 states $(\sigma_x^a \sigma_z^b \otimes \mathsf{Id}) \left|\phi^+\right\rangle$ for $a, b \in \{0,1\}$ on 2 qubits.

$$\mapsto |u \oplus r, v \oplus s\rangle_{\mathcal{W}} (\sigma_x(r)\sigma_z(s) \otimes \mathsf{Id})_{\mathcal{A}_1,\mathcal{A}_2} \left|\phi^+\right\rangle^{\otimes N}_{\mathcal{A}_1,\mathcal{A}_2}.$$

Finally, discard $\mathcal{W}$.

One can show that the resulting state is

$$\frac{1}{2^N} \sum_{r,s\in\{0,1\}^N} (\sigma_x(r)\sigma_z(s) \otimes \sigma_x(r)\sigma_z(s) \otimes \mathsf{Id}) |\psi\rangle_{\mathcal{H}} \left|\phi^+\right\rangle_{\mathcal{A}_1,\mathcal{A}_2} = \left|\phi^+\right\rangle_{\mathcal{H},\mathcal{A}_1} |\psi\rangle_{\mathcal{A}_2}, \tag{1}$$

where $|\psi\rangle$ is now "teleported" into the $\mathcal{A}_2$ register.

The Full Extractor. To generalize this idea to the setting where $|\psi\rangle \in \mathcal{H}$ is an arbitrary quantum state and $\{X_i, Z_i\}_i$ are an arbitrary collection of $2N$ observables, *we simply replace each* $\sigma_x(r)$ *and* $\sigma_z(s)$ *acting on* $\mathcal{H}$ *above with the corresponding parity observables* $X(r)$, $Z(s)$, defined analogously (for $r, s \in \{0,1\}^N$ as

$$Z(s) = \prod_{i=1}^{N} Z_i^{s_i} \quad \text{and} \quad X(r) = \prod_{i=1}^{N} X_i^{r_i}.$$

The rough intuition is that as long as the $\{X_i\}$ and $\{Z_i\}$ observables "behave like" Pauli observables with respect to $|\psi\rangle$, the resulting procedure will "teleport" $|\psi\rangle$ into the N-qubit register $\mathcal{A}_2$.

Relating Extracted State Measurements to Verifier Outputs. With the extracted state defined to be the state on $\mathcal{A}_2$ after performing the "generalized teleportation" described above, it remains to prove that the distribution $D_{P^*,\mathrm{Ext}}$ resulting from measuring the extracted state on $\mathcal{A}_2$ in the h-bases is indistinguishable from $D_{P^*,\mathsf{2\text{-}to\text{-}1}}$.

One can show (by a calculation) that $D_{P^*,\mathrm{Ext}}$ is the following distribution (differences from $D_{P^*,\mathsf{2\text{-}to\text{-}1}}$ in red)

1. Sample keys $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Cf}$. Run P^* to obtain $y, |\psi\rangle$.
2. For each i such that $h_i = 0$, measure the first bit of the prover's ith response register in the standard basis to obtain (and output) a bit b_i.
3. For each i such that $h_i = 1$, flip a random bit w_i and apply the unitary $Z_i^{w_i}$.
4. Measure $U|\psi\rangle$ in the Hadamard basis, obtaining strings $(d_1, \ldots, d_N)$.
5. For each i such that $h_i = 1$, compute (and output) $d_i \cdot (1, x_{0,i} \oplus x_{1,i}) \oplus w_i$.

We prove indistinguishability between the N-bit distributions $D_{P^*,\mathrm{Ext}}$ and $D_{P^*,\mathsf{2\text{-}to\text{-}1}}$ by considering N hybrid distributions, where the difference between Hybrid $j-1$ and Hybrid j is:

- an additional application of the unitary Z_j in Item 3, and
- an additional XOR of e_j (the jth standard basis vector) in Item 5.

To conclude the soundness proof, we show that Hybrid $j-1$ and Hybrid j in the following three steps.

- First, we prove that the marginal distributions of Hybrid $(j-1)$ and Hybrid j on $N \setminus \{j\}$ are indistinguishable due to the collapsing property of f_{pk_j}. Intuitively this holds because the marginal distributions on $N \setminus \{j\}$ only differ by the application of Z_j, which is undetectable by collapsing.
- By invoking an elementary lemma about N-bit indistinguishability, the task reduces to proving a 1-bit indistinguishability of the jth bit of Hybrid $(j-1)$ and Hybrid j, conditioned on an efficiently computable property of the marginal distributions on $N \setminus \{j\}$.
- Finally, we show that the indistinguishability of the jth bit holds due to the adaptive hardcore bit property of f_{pk_j}. At a very high level, the above jth bit property involves a measurement of X_j, and the two hybrids differ in whether a random Z_j^b is applied before X_j is measured; in words, this exactly captures the adaptive hardcore bit security game.
  We refer the reader to the full version for a complete proof of indistinguishability.

2.5 From a Verifier-Succinct Measurement Protocol to Succinct Arguments for BQP

Using Sections 2.3 and 2.4, we have constructed a *verifier-succinct* measurement protocol, for succinctly represented basis strings, with a single bit verifier challenge. What remains is to convert this into a (fully) succinct argument system for BQP (or QMA). This is accomplished via the following transformations:

- Converting a measurement protocol into a quantum verification protocol. As described earlier, this is achieved by combining the [FHM18] protocol for BQP verification with a limited quantum verifier (as modified by [ACGH20]) with our measurement protocol, using a PRF to generate a pseudorandom basis choice instead of a uniformly random basis choice for the [FHM18,ACGH20] verifier. This results in a verifier-succinct argument system for BQP/QMA with constant soundness error.
- Parallel repetition to reduce the soundness error. This follows from the "computational orthogonal projectors" property of the 1-bit challenge protocol and follows from [ACGH20] (we give a somewhat more abstract formulation of their idea in the full version). This results in a verifier-succinct argument system for BQP/QMA with negligible soundness error.
- Converting a verifier-succinct argument system into a fully succinct argument system. We elaborate on this last transformation below, as a few difficulties come up in this step.

Assume that we are given a (for simplicity, 4-message) verifier-succinct argument system for BQP/QMA. Let m_1, m_2, m_3, m_4 denote the four messages in such an argument system. In order to obtain a fully succinct argument system, we must reduce (1) the prover communication complexity $|m_2| + |m_4|$, and (2) the runtime of the verifier's decision predicate.

The first idea that comes to mind is to ask the prover to send short (e.g. Merkle tree) commitments σ_2 and σ_4 of m_2 and m_4, respectively, instead of

sending m_2 and m_4 directly. At the end of the interaction, the prover and verifier could then engage in a succinct interactive argument (of knowledge) for a (classical) NP statement that "the verifier would have accepted the committed messages underlying σ_2 and σ_4". One could potentially employ Kilian's succinct interactive argument of knowledge for NP which was recently shown to be post-quantum secure under the post-quantum LWE assumption [CMSZ21].

There are a few issues with this naive idea. First of all, the verifier's decision predicate is *private* (it depends on the secret key SK in the measurement protocol and the PRF seed for its basis), so the NP statement above is not well-formed. One reasonable solution to this issue is to simply have the verifier send this secret information st after the verifier-succinct protocol emulation has occurred and before the NP-succinct argument has started. For certain applications (e.g. obtaining a non-interactive protocol in the QROM) we would like to have a *public-coin* protocol; this can be achieved by using fully homomorphic encryption to encrypt this secret information in the *first* round rather than sending it in the clear in a later round. For this overview, we focus on the private-coin variant of the protocol.

Now, we can indeed write down the appropriate NP relation[12]

$$\begin{aligned}\mathcal{R}_V =\{((h, m_1, \sigma_2, m_3, \sigma_4, \mathsf{st}), (m_2, m_4)) : \sigma_2 = h(m_2) \text{ and} \\ \sigma_4 = h(m_4) \text{ and } V(\mathsf{st}, m_1, m_2, c, m_4) = \mathsf{accept}\}\end{aligned}$$

and execute the aforementioned strategy. However, this construction turns out not to work. Specifically, it does not seem possible to *convert* a cheating prover P^* in the above fully succinct protocol into a cheating prover P^{**} for the verifier-succinct protocol; for example, P^{**} needs to be able to produce a message m_2 given only m_1 from the verifier; meanwhile, the message m_1 can only be extracted from P^* by repeatedly rewinding P^*'s *last* message algorithm, which requires the verifier's secret information st as input! This does not correspond to a valid P^{**}, who does not have access to st when computing m_2.

Our refined compiler is to execute several arguments of knowledge: one right after the prover sends σ_2, proving knowledge of m_2; another one right after she sends σ_4, proving knowledge of m_4 (both before receiving the secret state st from the verifier); and a third one for the relation $\mathcal{R}_V$ described above. The first two arguments of knowledge are for the relation

$$\mathcal{R}_H = \{(h, \sigma), m) : h(m) = \sigma\}$$

This allows for *immediate extraction* of m_2 and m_3 and appears to clear the way for a reduction between the verifier-succinct and fully succinct protocol soundness properties.

However, there is one remaining problem: the argument-of-knowledge property of Kilian's protocol proved by [CMSZ21] is *insufficiently composable* to be used in our compiler. They demonstrate an extractor for Kilian's protocol that

[12] Note that the verifier also takes as input the QMA instance, but we suppress it here for clarity.

takes any quantum cheating prover that convinces the verifier and extracts a witness from them. However, their post-quantum extractor might significantly disturb the prover's state, meaning that once we extract m_2 above, we may not be able to continue the prover execution in our reduction.

Fortunately, a recent work [LMS21] shows that a slight variant of Kilian's protocol is a succinct argument of knowledge for NP satisfying a composable extraction property called "state-preservation." This security property is exactly what is required for our compiler to extract a valid cheating prover strategy P^{**} for the verifier-succinct argument given a cheating prover P^* for the compiled protocol. A complete discussion of this is given in the full version.

This completes our construction of a succinct argument system for BQP (and QMA). We discuss additional results (2-message protocols, zero knowledge, batch arguments) in the full version of this paper.

References

ACGH20. Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. pages 153–180, 2020. 4, 8, 14

ALM+92. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. pages 14–23, 1992. 2

AS92. Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; A new characterization of NP. pages 2–13, 1992. 2

BFL90. László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. pages 16–25, 1990. 2

BGKW88. Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. pages 113–131, 1988. 2

BV97. Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997. 2

CCY20. Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. pages 181–206, 2020. 4

CLLW20. Kai-Min Chung, Yi Lee, Han-Hsuan Lin, and Xiaodi Wu. Constant-round blind classical verification of quantum sampling. *arXiv preprint arXiv:2012.04848*, 2020. 2

CMSZ21. Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: breaking the quantum rewinding barrier. FOCS '21, 2021. 15

FHM18. Joseph F. Fitzsimons, Michal Hajdusek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Phys. Rev. Lett.*, 120:040501, Jan 2018. 3, 7, 14

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). pages 291–304, 1985. 2

GVW15. Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. pages 469–477, 2015. 5

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). pages 723–732, 1992. 2

LMS21. Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited (or: How to do quantum rewinding undetectably). Cryptology ePrint Archive, Report 2021/1543, 2021. https://ia.cr/2021/1543. 16

Mah18. Urmila Mahadev. Classical verification of quantum computations. pages 259–267, 2018. 2, 3, 4, 5, 7, 8, 9, 10

Mic94. Silvio Micali. A secure and efficient digital signature algorithm. Technical Memo MIT/LCS/TM-501b, Massachusetts Institute of Technology, Laboratory for Computer Science, April 1994. 2

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. pages 554–571, 2008. 5

PW08. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. pages 187–196, 2008. 5

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. pages 475–484, 2014. 8

Unr16. Dominique Unruh. Computationally binding quantum commitments. pages 497–527, 2016. 6

Vid20. Thomas Vidick. Interactions with quantum devices (course), 2020. http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf. 3, 10

VZ21. Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. pages 630–660, 2021. 7