Post-Quantum Multi-Party Computation

Amit Agarwal[1*], James Bartusek[†], Vipul Goyal[‡], Dakshita Khurana[1⋆], and Giulio Malavolta[§]

Abstract. We initiate the study of multi-party computation for classical functionalities in the plain model, with security against malicious quantum adversaries. We observe that existing techniques readily give a polynomial-round protocol, but our main result is a construction of *constant-round* post-quantum multi-party computation. We assume mildly super-polynomial quantum hardness of learning with errors (LWE), and quantum polynomial hardness of an LWE-based circular security assumption. Along the way, we develop the following cryptographic primitives that may be of independent interest:

- A spooky encryption scheme for relations computable by quantum circuits, from the quantum hardness of (a circular variant of) the LWE problem. This immediately yields the first quantum multi-key fully-homomorphic encryption scheme with classical keys.
- A constant-round post-quantum non-malleable commitment scheme, from the mildly super-polynomial quantum hardness of LWE.

To prove the security of our protocol, we develop a new straight-line non-black-box simulation technique against parallel sessions that does not clone the adversary's state. This technique may also be relevant to the classical setting.

1 Introduction

Secure multi-party computation (MPC) allows a set of parties to compute a joint function of their inputs, revealing only the output of the function while keeping their inputs private. General secure MPC, initiated in works such as [68, 33, 6, 14], has played a central role in modern theoretical cryptography. The last few years have seen tremendous research optimizing MPC in various ways, enabling a plethora of practical applications that include joint computations

[*]University of Illinois Urbana-Champaign. {amita2,dakshita}@illinois.edu. This material is based on work supported in part by DARPA under Contract No. HR001120C0024. 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.

[†]UC Berkeley. bartusek.james@gmail.com

[‡]NTT research and CMU. vipul@cmu.edu. Supported in part by the DARPA SIEVE program, NSF award 1916939, a gift from Ripple, a DoE NETL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cylab seed funding award.

[§]Max Planck Institute for Security and Privacy. giulio.malavolta@hotmail.it

on distributed medical data, privacy-preserving machine learning, e-voting, distributed key management, among others. The looming threat of quantum computers naturally motivates the problem of constructing protocols with *provable security against quantum adversaries*.

After Watrous' breakthrough work on zero-knowledge against quantum adversaries [64], the works of [20, 51, 41] considered variants of quantum-secure computation protocols, in the *two*-party setting. Very recently, Bitansky and Shmueli [10] obtained the first *constant-round* classical zero-knowledge arguments with security against quantum adversaries. Their techniques (and those of [1] in a concurrent work) are based on the recent non-black-box simulation technique of [8], who constructed two-message *classically-secure* weak zero-knowledge in the plain model. Unfortunately, it is unclear whether these protocols compose under parallel repetition. As a result, they become largely inapplicable to the constant-round multi-party setting.

There has also been substantial effort in constructing protocols for securely computing quantum circuits [25, 26, 23] (see Section 2.6 for further discussion). However, to the best of our knowledge, generic multi-party computation protocols with classical communication and security against quantum adversaries have only been studied in models with *trusted pre-processing or setup*. To make things even worse, [23] construct a maliciously-secure multi-party protocol for computing quantum ciruits, assuming the existence of a maliciously-secure post-quantum classical MPC protocol. This means that the only available implementations of such a building block require trusted pre-processing or a common reference string.

Post-Quantum MPC. In this work we initiate the study of MPC protocols that allow classical parties to securely compute general classical functionalities, and where security is guaranteed against *malicious quantum adversaries*. Our focus is on MPC in the *plain model*: Fully classical participants interact with each other with no access to trusted/pre-processed parameters or a common reference string. Multi-party protocols achieving security in these settings do not seem to have been previously analyzed in *any* number of rounds.

We stress that the challenges of proving post-quantum security of MPC protocols stretch far beyond the appropriate instantiations of the cryptographic building blocks (e.g. avoiding factoring- or discrete logarithm-based cryptosystems): Because quantum information behaves very differently from classical information, designing post-quantum protocols often requires new techniques to achieve provable security. As an example, a common strategy to prove classical security of MPC protocols is to define a simulator that can extract the inputs of the corrupted parties by "rewinding" them, i.e. taking a snapshot of the state of the adversary and split the protocol execution in multiple branches. However, when the adversary is a quantum machine, this technique becomes largely inapplicable since the no-cloning theorem (one of the fundamental principles of quantum mechanics) prevents us from creating two copies of an arbitrary quantum state. One of our key contributions is a new *parallel no-cloning non-black-box*

simulation technique that extends the work of [10], to achieve security against multiple parallel quantum verifiers.

1.1 Our Results

We begin by summarizing our main result: Classical multi-party computation with security against quantum circuits in the plain model. Here, parties communicate classically via authenticated point-to-point channels as well as broadcast channels, where everyone can send messages in the same round. In each round, all parties simultaneously exchange messages. The network is assumed to be synchronous with rushing adversaries, i.e. adversaries may generate their messages for any round after observing the messages of all honest parties in that round, but before observing the messages of honest parties in the next round. The (quantum) adversary may corrupt upto all but one of the participants. In this model, we obtain the following main result.

Theorem 1 (Informal). *Assuming mildly super-polynomial quantum hardness of LWE and AFS-spooky encryption for relations computable by polynomial-size quantum circuits, there exists a constant-round classical MPC protocol (in the plain model) maliciously secure against quantum polynomial-time adversaries.*

In more detail, our protocol is secure against any adversary $\mathsf{A} = \{\mathsf{A}_\lambda, \rho_\lambda\}_\lambda$, where each A_λ is the (classical) description of a polynomial-size quantum circuit and ρ_λ is some (possibly inefficiently computable) non-uniform quantum advice. Beyond being interesting in its own right, our plain-model protocol may serve as a useful stepping stone to obtaining interesting protocols for securely computing quantum circuits in the plain model, as evidenced by the work of [23]. This protocol is constructed in Sections 8 and 9 in the full version.

By "mildly" super-polynomial quantum hardness of LWE, we mean to assume that there exists a constant $c \in \mathbb{N}$, such that for large enough security parameter $\lambda \in \mathbb{N}$, no quantum polynomial time algorithm can distinguish LWE samples from uniform with advantage better than $\mathsf{negl}(\lambda^{\mathsf{ilog}(c,\lambda)})$, where $\mathsf{ilog}(c, \lambda)$ denotes the c-times iterated logarithm $\log\log\cdots_{c \text{ times}}(\lambda)$. We note that this is weaker than assuming the quasi-polynomial quantum hardness of LWE, i.e. the assumption that quantum polynomial-time adversaries cannot distinguish LWE samples from uniform with advantage better than $2^{-(\log\lambda)^c}$ for some constant $c > 1$.

A central technical ingredient of our work is an additive function sharing (AFS) spooky encryption scheme [21] for relations computable by quantum circuits. An AFS-spooky encryption scheme has a publicly-computable algorithm that, on input a set of ciphertexts $\mathsf{Enc}(\mathsf{pk}_1, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}_n, m_n)$ encrypted under *independently sampled* public keys and a (possibly quantum) circuit C, computes a new set of ciphertexts

$$\mathsf{Enc}(\mathsf{pk}_1, y_1), \ldots, \mathsf{Enc}(\mathsf{pk}_n, y_n) \ \textit{s.t.} \ \bigoplus_{i=1}^{n} y_i = C(m_1, \ldots, m_n).$$

In Section 4 in the full version we show how to construct AFS-spooky encryption for relations computable by quantum circuits, under an LWE-based circular security assumption. We refer the reader to Section 4.4 in the full version for the exact circular security assumption we need, which is similar to the one used in [52]. As a corollary, this immediately yields the first multi-key fully-homomorphic encryption [50] for quantum circuits with classical key generation and classical encryption of classical messages.

Theorem 2 (Informal). *Under an appropriate LWE-based circular security assumption, there exists an AFS-spooky encryption scheme for relations computable by polynomial-size quantum circuits with classical key generation and classical encryption of classical messages.*

Along the way to proving our main theorem, we construct and rely on constant-round zero-knowledge arguments against parallel quantum verifiers, and constant-round extractable commitments against parallel quantum committers. Parallel extractable commitments and zero-knowledge are formally constructed and analyzed in Sections 5 and 6 in the full version, respectively. We only show the construction of parallel extractable commitments in Section 3 in this paper. We point out that we do not obtain protocols that compose under *unbounded* parallel repetition. Instead we build a bounded variant (that we also refer to as multi-verifier zero-knowledge and multi-committer extractable commitments) that suffices for our applications.

Theorem 3 (Informal). *Assuming the quantum polynomial hardness of LWE and the existence of AFS-spooky encryption for relations computable by polynomial-size quantum circuits, there exists:*

- *A constant-round classical argument for NP that is computational-zero-knowledge against parallel quantum polynomial-size verifiers.*
- *A constant-round classical commitment that is extractable against parallel quantum polynomial-size committers.*

In addition, we initiate the study of post-quantum non-malleable commitments. Specifically, we construct and rely on constant-round post-quantum non-malleable commitments based on the super-polynomial hardness assumption described above. The formal construction and analysis can be found in Section 7 in the full version.

Theorem 4 (Informal). *Assuming the mildly super-polynomial quantum hardness of LWE and the existence of fully-homomorphic encryption for quantum circuits, there exists a constant-round non-malleable commitment scheme secure against quantum polynomial-size adversaries.*

We also obtain quantum-secure non-malleable commitments in $O(\mathsf{ilog}(c, \lambda))$ rounds for any constant $c \in \mathbb{N}$ based on any quantum-secure extractable commitment. In particular, plugging in these commitments instead of our constant round non-malleable commitments gives an $O(\mathsf{ilog}(c, \lambda))$ round quantum-secure MPC from any quantum AFS-spooky encryption scheme.

2 Technical Overview

2.1 Background

Our starting point is any constant-round post-quantum MPC protocol maliciously secure in the programmable common reference string (CRS) model. Such a protocol can be obtained, for example, based on the semi-maliciously secure MPC protocols of [2, 53] in the CRS model. Specifically, assuming the existence of post-quantum zero-knowledge in the CRS model (that can be obtained based on the quantum hardness of LWE [60]) and the quantum hardness of LWE, these works obtain multi-party computation for classical circuits in the CRS model with the following property: There exists an ideal-world simulator that *programs the CRS*, interacts in a straight-line, black-box manner with any quantum adversary corrupting an arbitrary subset of the players, and outputs a view that is indistinguishable from the real view of the adversary, including the output of honest parties.

Thus, a natural approach to achieving post-quantum MPC in the plain model is to then securely implement a multi-party functionality that generates the aforementioned CRS. Specifically, we would like a set of n parties to jointly execute a *coin-flipping protocol*. Such a protocol outputs a uniformly random string that may then be used to implement post-quantum secure MPC according to [2, 53]. The programmability requirement on the CRS roughly translates to ensuring that for any quantum adversary, there exists a simulator that on input a random string s, can force the output of the coin-flipping protocol to be equal to s. A protocol satisfying this property is often referred to as a *fully-simulatable* multi-party coin-flipping protocol.

Post-Quantum Multi-Party Coin-Flipping. Existing constant-round protocols [65, 36] for multi-party coin-flipping against classical adversaries make use of the following template. Each participant first commits to a uniformly random string using an appropriate perfectly binding commitment.[1] In a later phase, all participants reveal the values they committed to, without actually revealing the randomness used for commitment. Additionally, each participant proves (in zero-knowledge) to every other participant that they opened to the same value that they originally committed to. If all zero-knowledge arguments verify, the protocol output is computed as the sum of the openings of all participants. To highlight challenges to construct constant-round protocols, we elaborate on this template and outline a simple polynomial-round coin tossing protocol. Readers familiar with this template for multi-party coin-tossing may skip to the next page.

A Simple Protocol in Polynomially Many Rounds. In order to motivate the challenges involved in constructing a post-quantum *constant-round* multi-party coin tossing protocol, we first outline a simple protocol that requires *polynomially many* rounds, and follows from ideas in existing work. Our starting

[1] We actually require this commitment to also satisfy a property called *non-malleability*, which we discuss later in this section.

point is the polynomial-round post-quantum zero-knowledge protocol due to Watrous [64]. Ideas developed in [10] can be used almost immediately to convert this to a post-quantum extractable commitment scheme, assuming polynomial hardness of LWE (or, more generally, any post-quantum oblivious transfer). For completeness, we outline how this is done in Appendix A in the full version. Next, it is possible to use the resulting post-quantum secure extractable commitment to obtain post-quantum multiparty fully-simulatable coin flipping, that admits a straight-line simulator in the dishonest majority setting. The protocol requires rounds that grow linearly with the number of parties and polynomially with the security parameter, as described in Figure 1.

n-Party Coin tossing

Common input: 1^λ, 1^n.

1. For each $i \in [n]$, party P_i samples $r_i \leftarrow \{0,1\}^\lambda$.
2. Sequentially, for every $i \in [n]$, $j \in [n] \setminus \{i\}$ parties P_i, P_j execute a post-quantum extractable commitment where P_i commits to r_i and P_j is the receiver.
3. P_i broadcasts r_i.
4. For every $i \in [n]$, $j \in [n] \setminus \{i\}$ parties P_i, P_j *sequentially* execute a post-quantum ZK protocol where P_i is the prover and P_j is the verifier. P_i proves to P_j (in zero-knowledge) that the value committed via the extractable commitment (in Step 2) is consistent with the value broadcasted (in Step 3).
5. If all the proofs where P_i is verifier are accepting, P_i outputs $\bigoplus_{i=1}^{n} r_i$.

Fig. 1: Multiparty Coin Tossing

Recall that the simulator Sim of any coin-flipping protocol will obtain a uniformly random string r^* from the ideal functionality, and must force this value as the output. The Sim for the protocol in Figure 1 samples r_i uniformly at random on behalf of each honest party P_i, and commits to r_i in Step 2 following honest sender strategy. At the same time, Sim runs Ext to (sequentially) extract the value committed by every corrupted party in Step 2. This allows the simulator to compute $\bigoplus_{i\in\mathbb{M}} r_i$, where $\mathbb{M}$ denotes the set of corrupted parties. In Step 3, the simulator broadcasts values r'_i on behalf of honest parties such that $\bigoplus_{i\in[n]\setminus\mathbb{M}} r'_i = \bigoplus_{i\in\mathbb{M}} r_i \oplus r^*$. Finally, it invokes the simulator of the ZK protocol to produce proofs on behalf of honest parties. It is easy to see that the output would indeed end up being the intended output r^*.

Notice that replacing Watrous' polynomial-round ZK protocol with the constant-round ZK of [10, 1] only decreases the rounds to linear in the number of parties. To decrease the number of rounds to constant, it is clear that one would need to find a way to execute the commitment sessions (Step 2) and ZK sessions (Step 4) in parallel. While the recent work of Bitansky and Shmueli [10] builds constant-round post-quantum zero-knowledge, their protocol and its guarantees turn out

to be insufficient for the parallel setting. In this setting, a single prover would typically need to interact in parallel with $(n-1)$ different verifiers, a subset or all of which may be adversarial. It should be possible for a simulator to *simultaneously* simulate the view of multiple parallel verifiers. In addition, the argument should continue to satisfy soundness, even if a subset of verifiers colludes with a (cheating) prover.

Post-Quantum Parallel Zero-Knowledge. We overcome this barrier by building the first constant-round zero-knowledge argument secure against *parallel quantum verifiers* from quantum polynomial hardness of an LWE-based circular security assumption. This improves upon the work of [10, 1] who provided arguments with provable security only against a single quantum verifier. Very roughly, the approach in [10, 1] relies on a modification of the [8] homomorphic trapdoors paradigm. We do not assume familiarity with the details of this protocol or paradigm, and will in fact discuss a (variant of) this in the next subsection. For now, we simply point out that in this paradigm, the verifier generates an initial FHE ciphertext and public key, as well as some additional information to enable simulation. The simulator *homomorphically evaluates* the verifier's (quantum) circuit over the initial FHE ciphertext and then uses the result of this evaluation to recover secrets that will enable simulation.

However, when a prover interacts with several verifiers at once, each verifier will generate its own FHE ciphertexts. In a nutshell, in the parallel setting the simulator can no longer perform individual homomorphic evaluations corresponding to each verifier, due to no-cloning. To address this issue, we develop a novel **parallel no-cloning** simulation strategy. This strategy relies on a novel technique that enables the simulator to *peel away* secret keys of this FHE scheme layer-by-layer. An overview of this technique can be found in Section 2.2.

Our technique also crucially relies on a strong variant of quantum fully-homomorphic encryption that allows for homomorphic operations under multiple keys at once. The encryption scheme that we use is a quantum generalization of the notion of additive function sharing (AFS) *spooky encryption* [21]. As a contribution of independent interest, we build the first AFS-spooky encryption (that also implies multi-key FHE) for quantum circuits from a circular variant of the LWE assumption. We give an overview of our construction in Section 2.3.

Post-Quantum Non-malleable Commitments. Our construction of zero-knowledge against parallel quantum verifiers gives rise to a coin-flipping protocol that is secure as long as at least one participant is honest, and all committed strings are independent of each other. However, ensuring such independence is not straightforward, even in the classical setting. In fact, upon seeing an honest party's commitment string c, a malicious, rushing adversary may be able to produce a string c' that commits to a related message. This is known as a malleability attack, and can be prevented by relying on *non-malleable commitments*. In this work, we devise the first post-quantum non-malleable commitments based on slightly superpolynomial hardness of LWE. An overview of our construction can be found in Section 2.4.

Finally, we discuss how to combine all these primitives to build our desired coin-tossing protocol, and a few additional subtleties that come up in the process, in Section 2.5.

2.2 A New Parallel No-Cloning Non-Black-Box Simulation Technique

In the following we give a high-level overview of our constant-round zero-knowledge protocol secure against parallel quantum verifiers. In favor of a simpler exposition, we first describe a *parallel extractable commitment* protocol. A parallel extractable commitment is a commitment where a single receiver interacts in parallel with multiple committers, each committing to its own independent message. The main challenge in this setting is to simulate the view of an adversary corrupting several of these committers, while *simultaneously* recovering all committed messages. Once we build a parallel extractable commitment, obtaining a parallel zero-knowledge protocol becomes a simple exercise (that we discuss towards the end of this overview).

Throughout the following overview we only consider adversaries that are (i) *non-aborting*, i.e. they never interrupt the execution of the protocol, and (ii) *explainable*, i.e. their messages always lie in the support of honestly generated messages, though they can select their random coins and inputs arbitrarily. We further simplify our overview by only considering (iii) *classical* adversaries, while being mindful to avoid any kind of state cloning during extraction. In the end of this overview we discuss how to remove these simplifications.

Cryptographic Building Blocks. Before delving into the description of our protocol, we introduce the technical tools needed for our construction. A fully-homomorphic encryption (FHE) scheme [29] allows one to compute any function (in its circuit representation) over some encrypted message $\mathsf{Enc}(\mathsf{pk}, m)$, without the need to decrypt it first. We say that an FHE is multi-key [50] if it supports the homomorphic evaluation of circuits even over messages encrypted under *independently sampled* public keys:

$$\{\mathsf{Enc}(\mathsf{pk}_i, m_i)\}_{i\in[n]} \xrightarrow{\mathsf{Eval}((\mathsf{pk}_1,\ldots,\mathsf{pk}_n),C,\cdot)} \mathsf{Enc}((\mathsf{pk}_1,\ldots,\mathsf{pk}_n), C(m_1,\ldots,m_n)).$$

Clearly, decrypting the resulting ciphertext should require the knowledge of all of the corresponding secret keys $(\mathsf{sk}_1,\ldots,\mathsf{sk}_n)$. Other than semantic security, we require that the scheme is compact, in the sense that the size of the evaluated ciphertext is proportional to $|C(m_1,\ldots,m_n)|$ (and possibly the number of parties n) but does not otherwise depend on the size of C.

The second tool that we use is compute and compare obfuscation [66, 35]. A compute and compare program $\mathbf{CC}[f,u,z]$ program is defined by a function f, a lock value u, and an output z. On input a string x, the program returns z if and only if $f(x) = u$. The obfuscator Obf is guaranteed to return an obfuscated program $\widetilde{\mathbf{CC}}$ that is indistinguishable from a program that rejects any input, as long as u has sufficient entropy conditioned on f and z. Finally, we use a

conditional disclosure of secret $(\mathrm{CDS})^2$ scheme. Recall that this is an interactive protocol parametrized by an NP relation $\mathcal{R}$ where both the sender and the receiver share a statement x and in addition, the sender has a secret message m. At the end of the interaction, the receiver obtains m if and only if it knows a valid witness w such that $\mathcal{R}(x, w) = 1$.

A Strawman Solution. We now describe a naive extension of the [10, 1] approach to the parallel setting (where a receiver interacts with multiple committers), and highlight its pitfalls. We do not assume familiarity with [10, 1]. To commit to messages $(m_1, \ldots, m_n)$, the committers and the receiver engage in the following protocol.

- Each committer samples a key pair of a multi-key FHE scheme $(\mathsf{pk}_i, \mathsf{sk}_i)$, a uniform trapdoor td_i, and a uniform lock value lk_i, and sends to the receiver:
  1. A commitment $\mathsf{c}_i = \mathsf{Com}(\mathsf{td}_i)$.
  2. An FHE encryption $\mathsf{Enc}(\mathsf{pk}_i, \mathsf{td}_i)$.
  3. An obfuscation $\widetilde{\mathbf{CC}}_i$ of the program $\mathbf{CC}[\mathsf{Dec}(\mathsf{sk}_i, \cdot), \mathsf{lk}_i, (\mathsf{sk}_i, m_i)]$.
- The receiver engages each committer in a (parallel) execution of a CDS protocol where the i'th committer sends lk_i if the receiver correctly guesses a valid pre-image of c_i.

At a high level, the fact that the protocol hides the message m_i is ensured by the following argument. Since the receiver cannot invert c_i, it cannot guess td_i and therefore the CDS protocol will return 0. This in turn means that the lock lk_i is hidden from the receiver, and consequently that the obfuscated program is indistinguishable from a null program. This is, of course, an informal explanation, and we refer the reader to [8, 10, 1] for a formal security analysis.

We now turn to the description of the extractor. The high-level strategy is the following: Upon receiving the first message from all committers, the extractor uses the FHE encryption $\mathsf{Enc}(\mathsf{pk}_i, \mathsf{td}_i)$ and the code of the adversary to run the CDS protocol homomorphically (on input td_i) to recover an FHE encryption of lk_i. Then the extractor feeds it as an input to the obfuscated program $\widetilde{\mathbf{CC}}_i$, which returns (sk_i, m_i).

Unfortunately this approach has a major limitation: It implicitly assumes that each corrupted party is a local algorithm. In other words, we are assuming that the adversary consists of individual subroutines (one per corrupted party), which may not necessarily be the case. As an example, if the adversary were to somehow implement a strategy where corrupted machines do not respond until *all* receiver messages have been delivered, then the above homomorphic evaluation would get stuck and return no output. It is also worth mentioning that what makes the problem challenging is our inability to clone the state of the adversary. If we were allowed to clone its state, then we could extract messages one by one, by running a separate thread under each FHE key.

[2] In the body of the paper we actually resort to a slightly stronger tool, namely a secure function evaluation protocol with statistical circuit privacy.

Multi-Key Evaluation. A natural solution to circumvent the above issue is to rely on multi-key FHE evaluation. Using this additional property, the extractor can turn the ciphertexts $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{td}_1), \ldots, \mathsf{Enc}(\mathsf{pk}_n, \mathsf{td}_n)$ into a single encryption

$$\mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), (\mathsf{td}_1, \ldots, \mathsf{td}_n))$$

under the hood of all public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. Given this information, the extractor can homomorphically evaluate all instances of the CDS protocol at once, using the code of the adversary, no matter how intricate. This procedure allows the extractor to obtain the encryption of each lock value $\mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), \mathsf{lk}_i)$. In the single committer setting, we could then feed this into the corresponding obfuscated program and call it a day.

However, in the parallel setting, even given multi-key FHE, it is unclear how to proceed. If the compute and compare program $\widetilde{\mathbf{CC}}_i$ tried to decrypt such a ciphertext, it would obtain (at best) an encryption under the remaining public keys. Glossing over the fact that the structure of single-key and multi-key ciphertexts might be incompatible, it is unlikely that

$$\mathsf{Dec}(\mathsf{sk}_i, \mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), \mathsf{lk}_i)) = \mathsf{lk}_i$$

which is what we would need to trigger the compute and compare program. The general problem here is that each compute and compare program cannot encode information about other secret keys, thus making it infeasible to decrypt multi-key ciphertexts. One approach to resolve this issue would be to ask all committers to jointly obfuscate a compute and compare program that encodes all secret keys at once. However, this seems to require a general-purpose MPC protocol, which is what we are trying to build in the first place. Therefore, we outline a different approach by imagining a special kind of multi-key fully homomorphic encryption scheme.

A spooky encryption[3] scheme [21] is an FHE scheme that supports a special *spooky evaluation* algorithm, that generates no-signaling correlations among independently encrypted messages. We will restrict attention to a sub-class of no-signaling relations called *additive function sharing* (AFS) relations, and we will call the scheme AFS-spooky. More concretely, on input a circuit C and n independently generated ciphertexts (under independently generated public keys), the algorithm $\mathsf{Spooky.Eval}$ produces

$$\{\mathsf{Enc}(\mathsf{pk}_i, m_i)\}_{i\in[n]} \xrightarrow{\mathsf{Spooky.Eval}((\mathsf{pk}_1,\ldots,\mathsf{pk}_n),C,\cdot)} \{\mathsf{Enc}(\mathsf{pk}_i, y_i)\}_{i\in[n]} \ \ s.t. \ \bigoplus_{i=1}^{n} y_i = C(m_1, \ldots, m_n).$$

It is not hard to see that AFS-spooky encryption is a special case of multi-key FHE where multi-key ciphertexts have the following structure

$$\mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), m) = \{\mathsf{Enc}(\mathsf{pk}_i, y_i)\}_{i\in[n]} \ \ s.t. \ \bigoplus_{i=1}^{n} y_i = m.$$

[3] As a historical remark, while the name is inspired by Einstein's quote "spooky action at a distance" referring to entangled quantum states, the concept of spooky encryption (as defined in [21]) is entirely classical.

This additional structure is going to be our main leverage for constructing an efficient extractor.

The Extractor. Going back to our extractor, our next technical insight is to look for a mechanism to *peel away* encryption layers one by one from an AFS-spooky (multi-key) ciphertext. Our extractor will achieve this via careful *homomorphic* evaluation of the independently generated programs $(\widetilde{\mathbf{CC}}_1, \ldots, \widetilde{\mathbf{CC}}_n)$, as described below.

- First, homomorphically execute the code of the adversary using the AFS-spooky scheme to obtain

$$\mathsf{ct}_1 = \mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), \mathsf{lk}_1), \ldots, \mathsf{ct}_n = \mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), \mathsf{lk}_n),$$

  as described above.
- Parse ct_n as a collection of individual ciphertexts

$$\mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_n), \mathsf{lk}_n) = \{\mathsf{Enc}(\mathsf{pk}_i, y_i)\}_{i\in[n]} = \{\mathsf{Enc}(\mathsf{pk}_i, y_i)\}_{i\in[n-1]} \cup \underbrace{\{\mathsf{Enc}(\mathsf{pk}_n, y_n)\}}_{\tilde{\mathsf{ct}}_n}.$$

  Note that we can interpret the first $n-1$ elements as an AFS-spooky ciphertext encrypted under $(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1})$:

$$\tilde{\mathsf{ct}} = \{\mathsf{Enc}(\mathsf{pk}_i, y_i)\}_{i\in[n-1]} = \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \bigoplus_{i=1}^{n-1} y_i\right) = \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \tilde{y}\right)$$

  where $\tilde{y} = \bigoplus_{i=1}^{n-1} y_i$.
- Let Γ be the following function

$$\Gamma(\zeta) : \mathsf{Spooky.Eval}(\mathsf{pk}_n, \zeta \oplus \cdot, \tilde{\mathsf{ct}}_n)$$

  which homomorphically computes the XOR of ζ with the plaintext of $\tilde{\mathsf{ct}}_n$. Compute the following nested AFS-spooky correlation

$$\begin{aligned}
\widehat{\mathsf{ct}} &= \mathsf{Spooky.Eval}((\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}), \Gamma, \tilde{\mathsf{ct}}) \\
&= \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \mathsf{Spooky.Eval}(\mathsf{pk}_n, \tilde{y} \oplus \cdot, \tilde{\mathsf{ct}}_n)\right) && (1) \\
&= \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \mathsf{Enc}\left(\mathsf{pk}_n, \bigoplus_{i=1}^{n} y_i\right)\right) && (2) \\
&= \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \mathsf{Enc}\left(\mathsf{pk}_n, \mathsf{lk}_n\right)\right) && (3)
\end{aligned}$$

  by interpreting $\tilde{\mathsf{ct}}_n$ as a single key ciphertext. Here (1) follows by substituting Γ, and (2) follows by correctness of the AFS-spooky evaluation.
- Run the obfuscated compute and compare program homomorphically to obtain an encryption of sk_n and m_n under $(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1})$

$$\begin{aligned}
\mathsf{Spooky.Eval}\left((\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}), \widetilde{\mathbf{CC}}_n, \widehat{\mathsf{ct}}\right) &= \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \widetilde{\mathbf{CC}}_n\left(\mathsf{Enc}\left(\mathsf{pk}_n, \mathsf{lk}_n\right)\right)\right) \\
&= \mathsf{Enc}\left(\left(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}\right), \left(\mathsf{sk}_n, m_n\right)\right).
\end{aligned}$$

– Using the encryption of sk_n under $(\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1})$, update the initial ciphertexts $(\mathsf{ct}_1, \ldots, \mathsf{ct}_{n-1})$ by homomorphically decrypting their last component and adding the resulting string. This allows the extractor to obtain

$$\mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}), \mathsf{lk}_1), \ldots, \mathsf{Enc}((\mathsf{pk}_1, \ldots, \mathsf{pk}_{n-1}), \mathsf{lk}_{n-1}).$$

– Recursively apply the procedure described above until $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{lk}_1)$ is recovered, then feed this ciphertext as an input to $\widetilde{\mathbf{CC}}_1$ to obtain (sk_1, m_1) in the clear. Iteratively recover $(\mathsf{sk}_2, \ldots, \mathsf{sk}_n)$ by decrypting the corresponding ciphertexts. At this point the extractor knows all secret keys and can decrypt the transcript of the interaction together with the committed messages.

To summarize, this extractor will isolate single-key ciphertexts (albeit in a nested form) by relying on AFS-spooky encryption. These ciphertexts by design will be compatible with compute and compare programs. In turn, evaluating the program *under the encryption* allows us to *escape* from the newly introduced layer. Repeating this procedure recursively eventually leads to a complete recovery of the plaintexts.

We stress that, although the extraction algorithm repeats the nesting operation n times, the additional encryption layer introduced in each iteration is immediately peeled off by executing the obfuscated compute and compare program. Thus the above procedure runs in (strict) polynomial time for *any polynomial* number of parties n.

Parallel Zero Knowledge. The above outline is deliberately simplified and ignores some subtle issues that arise during the analysis of the protocol. As an example, we need to ensure that the adversary is not able to *maul* the commitment of the trapdoor into a CDS encryption to be used in the CDS protocol. This issue also arose in [10], and we follow their approach of using non-uniformity in a reduction to the semantic security of the quantum FHE scheme. [10] also present the technical tools needed to lift the protocol to the setting of malicious and possibly aborting adversaries (as opposed to explainable), and we roughly follow their approach. However, it is worth pointing out that [10] directly construct a zero-knowledge argument, without first constructing and analyzing a stand-alone extractable commitment. Since we use a parallel extractable commitment as a building block in the our coin-flipping protocol, we analyze the above as a stand-alone commitment, which requires a few modifications to the protocol and proof techniques. More discussion about this can be found in Section 3.

Now, we describe how to obtain parallel zero-knowledge (i.e. zero-knowledge against multiple verifiers) from parallel extractable commitment. This is accomplished in a routine manner by enhancing a standard Σ protocol with a stage where each verifier commits to its Σ protocol challenge using a parallel extractable commitment. Using the extractor, the simulator can obtain the challenges ahead of time and can therefore simulate the rest of the transcript, without the need to perform state cloning.

It remains to argue that our extraction strategy does not break down in the presence of quantum adversaries. Observe that the only step that involves

the execution of a quantum circuit is the AFS-spooky evaluation of the CDS protocol, under the hood of $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$. Assuming that we can construct AFS-spooky encryption for relations computable by quantum circuits (which we show in Section 2.3), the remainder of the extraction algorithm only depends on the encryptions of $(\mathsf{lk}_1, \ldots, \mathsf{lk}_n)$, which are classical strings. Once the extractor recovers all the secret keys, it can decrypt the (possibly quantum) state of the adversary resulting from the homomorphic evaluation of the CDS, and resume the protocol execution, without the need to clone the adversary's state.

2.3 Quantum AFS-Spooky Encryption

We now turn to the construction of AFS-spooky encryption for relations computable by quantum circuits. The main technical contribution of this section is a construction of multi-key fully-homomorphic encryption for quantum circuits with classical key generation and classical encryption of classical messages. Such schemes were already known in the *single*-key setting, due to [52, 11].

Background. At a very high level, these single-key schemes follow a paradigm introduced by Broadbent and Jeffery [13], which makes use of the quantum one-time pad (QOTP). The QOTP is a method of perfectly encrypting arbitrary quantum states with a key that consists of only classical bits. [13] suggest to encrypt a quantum state with a quantum one-time pad (QOTP), and then encrypt the classical bits that comprise the QOTP using a classical fully-homomorphic encryption scheme. One can then apply quantum gates to the encrypted quantum state, and update the classical encryption of the one-time pad appropriately. A key feature of this encryption procedure is that while an encryption of a quantum state necessarily must be a quantum state, an encryption of classical information does not necessarily have to include a quantum state. Indeed, one can simply give a classical one-time pad encryption of the data, along with a classical fully-homomorphic encryption of the pad.

However, the original schemes presented by Broadbent and Jeffery [13] and subsequent work [24] based on their paradigm left much to be desired. In particular, they required even a classical encryptor to supply quantum "gadgets" encoding their secret key. These gadgets were then used to evaluate a particular non-Clifford gate over encrypted data.[4] The main innovation in the work of [52] was to remove the need for quantum gadgets, instead showing how to evaluate an appropriate non-Clifford gate using just *classical* information supplied by the encryptor.

Encrypted CNOT Operation. In more detail, evaluating a non-Clifford gate on a ciphertext $(\mathsf{ct}, |\phi\rangle)$, where ct is an FHE encryption of a QOTP key and $|\phi\rangle$ is a quantum state encrypted under the QOTP key, involves an operation (referred to as encrypted CNOT) that somehow must "teleport" the bits encrypted in ct

[4] We also remark here that [34] presented a *multi*-key scheme based on this paradigm, but with the same drawbacks. Note that compactness and classical encryption are crucial in our setting, as per the discussion in the previous section.

into the state $|\phi\rangle$. [52] gave a method for doing this, as long as the ciphertext ct is encrypted under a scheme with some particular properties. Roughly, the scheme must support a "natural" XOR homomorphic operation, it must be circuit private with respect to this homomorphism, and perhaps most stringently, there must exist some trapdoor that can be used to recover the message and the *randomness* used to produce any ciphertext.

[52] observed that the dual-Regev encryption scheme [30] (with large enough modulus-to-noise ratio) does in fact satisfy these properties, as long as one generates the public key matrix $\mathbf{A}$ along with a trapdoor. However, recall that ct was supposed to be encrypted under a fully-homomorphic encryption scheme. [52] resolves this by observing that ciphertexts encrypted under the dual variant of the [31] fully-homomorphic encryption scheme actually already contain a dual-Regev ciphertext. In particular, a dual-GSW ciphertext encrypting a bit μ is a matrix $\mathbf{M} = \mathbf{AS} + \mathbf{E} + \mu\mathbf{G}$, where $\mathbf{G}$ is the gadget matrix. The final column of $\mathbf{M}$ is $\mathbf{As} + \mathbf{e} + \mu[0, \ldots, 0, q/2]^\top$, which is exactly a dual-Regev ciphertext encrypting μ under public key $\mathbf{A}$. Note that, crucially, if the dual-GSW public key $\mathbf{A}$ is drawn with a trapdoor, then this trapdoor also functions as a trapdoor for the dual-Regev ciphertext. Thus, an evaulator can indeed perform the encrypted CNOT operation on any ciphertext $(\mathsf{ct}, |\phi\rangle)$, by first extracting a dual-Regev ciphertext ct' from ct and then proceeding.

Challenges in the Multi-Key Setting. Now, it is natural to ask whether this approach readily extends to the multi-key setting. Namely, does there exist a multi-key FHE scheme where any (multi-key) ciphertext contains within it a dual-Regev ciphertext *with a corresponding trapdoor*? Unfortunately, this appears to be much less straightforward than in the single-key setting, for the following reason. Observe that (dual) GSW homomorphic operations over ciphertexts $\mathbf{M}_i = \mathbf{AS}_i + \mathbf{E}_i + \mu_i\mathbf{G}$ always maintain the same $\mathbf{A}$ matrix, while updating $\mathbf{S}_i$, $\mathbf{E}_i$, and μ_i. Thus, a trapdoor for $\mathbf{A}$ naturally functions as a trapdoor for the dual-Regev ciphertext that consitutes the last column of $\mathbf{M}_i$. However, LWE-based multi-key FHE schemes from the literature [18, 53, 59, 12] include a *ciphertext expansion* procedure, which allows an evaluator, given public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_n$, and a ciphertext ct encrypted under some pk_i, to convert ct into a ciphertext $\hat{\mathsf{ct}}$ encrypted under all keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_n$. Now, even if these public keys are indeed matrices $\mathbf{A}_1, \ldots, \mathbf{A}_n$ drawn with trapdoors $\tau_1, \ldots, \tau_n$, it is unclear how to combine $\tau_1, \ldots, \tau_n$ to produce a trapdoor $\hat{\tau}$ for the "expanded" ciphertext. Indeed, the expanded ciphertext generally can no longer be written as some $\mathbf{AS} + \mathbf{E} + \mu\mathbf{G}$, since the expansion procedure constructs a highly structured matrix that includes components from the ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$, as well as auxiliary encryptions of the randomness used to produce the ciphertexts (see e.g. [53]).

A Solution Based on Key-Switching. Thus, we take a different approach. Rather than attempting to tweak known ciphertext expansion procedures to also support "trapdoor expansion", we rely on the notion of key-switching, which is a method of taking a ciphertext encrypted under one scheme and converting it into a ciphertext encrypted under another scheme. The observation, roughly, is that

we do not need to explicitly maintain a trapdoor for the multi-key FHE scheme, as long as it is possible to convert a multi-key FHE ciphertext into a dual-Regev ciphertext that *does* explicitly have a trapdoor. In fact, we will consider a natural multi-key generalization of dual-Regev, as described below. Key switching is possible as long as the second scheme has sufficient homomorphic properties, namely, it can support homomorphic evaluation of the *decryption circuit* of the first scheme.

Fortunately, the dual-Regev scheme is already linearly homomorphic, and many known classical multi-key FHE schemes [18, 53, 59, 12] support *nearly linear decryption*, which means that decrypting a ciphertext simply consists of applying a linear function (derived from the secret key) and then rounding. Thus, as long as the evaluator has the secret key of the multi-key FHE ciphertext encrypted under a dual-Regev public key with a trapdoor, they can first key-switch the multi-key FHE ciphertext ct into a dual-Regev ciphertext ct', and then proceed with the encrypted CNOT operation.

It remains to show how an evaluator may have access to such a dual-Regev encryption. Since we are still in the multi-key setting, we will need a ciphertext and corresponding trapdoor expansion procedure for dual-Regev. However, we show that such a procedure is much easier to come by when the scheme only needs to support *linear* homomorphism (as is the case for the dual-Regev scheme) rather than *full* homomorphism. Each party can draw its own dual-Regev public key $\mathbf{A}_i$ along with a trapdoor τ_i, and encrypt its multi-key FHE secret key under $\mathbf{A}_i$ to produce a ciphertext ct_i. The evaluator can then treat the block-diagonal matrix $\hat{\mathbf{A}} = \mathsf{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ as an "expanded" public key.[5] Now, the message and randomness used to generate a ciphertext encrypted under $\hat{\mathbf{A}}$ may be recovered by applying τ_1 to the first set of entries of the ciphertext, applying τ_2 to the second set of entries and so on. This observation, combined with an appropriate expansion procedure for the ciphertexts ct_i, allows an evaluator to convert any multi-key FHE ciphertext into a multi-key dual-Regev ciphertext with trapdoor. Given a classical multi-key FHE scheme with nearly linear decryption, this suffices to build multi-key quantum FHE with classical key generation and encryption.

Distributed Setup. We showed above how to convert any classical multi-key FHE scheme into a quantum multi-key FHE scheme, as long as the classical scheme has nearly linear decryption. However, most LWE-based classical multi-key FHE schemes operate in the common random string (CRS) model, which assumes that all parties have access to a common source of randomness, generated by a trusted party. Thinking back to our application to parallel extractable commitments, it is clear that this will not suffice, since we have no CRS a priori, and a receiver that generates a CRS maliciously may be able to break hiding of the scheme. Thus, we rely on the multi-key FHE scheme of [12], where instead of assuming a CRS, the parties participate in a distributed setup procedure. In particular, each party (and in our application, each committer) generates some

[5] Actually this expansion should be done slightly more carefully, see Section 4.4 in the full version for details.

public parameters pp_i, which are then combined publicly to produce a single set of public parameters pp, which can be used by anyone to generate their own public key / secret key pair.

This form of distributed setup indeed suffices to prove the hiding of our parallel commitment, so it remains to show that our approach, combined with [12], yields a quantum multi-key FHE scheme with distributed setup. First, the [12] scheme does indeed enjoy nearly linear decryption, so plugging it into our compiler described above gives a functional quantum multi-key FHE scheme. Next, we need to confirm that our compiler does not destroy the distributed setup property. This follows since each party draws its own dual-Regev public key with trapdoor without relying on any CRS, or even any public parameters.

Quantum AFS-Spooky Encryption. Finally, we show, via another application of key-switching, how to construct a quantum AFS-spooky encryption scheme (with distributed setup). Recall that we only require "spooky" interactions to hold over classical ciphertexts. That is, for any *quantum* circuit C with classical outputs, given ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_n$ encrypting $|\phi_1\rangle, \ldots, |\phi_n\rangle$ respectively under public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_n$, an evaluator can produce ciphertexts $\mathsf{ct}'_1, \ldots, \mathsf{ct}'_n$ where ct'_i encrypts y_i under pk_i, and such that $\bigoplus_{i=1}^{n} y_i = C(|\phi_1\rangle, \ldots, |\phi_n\rangle)$.

Now, using our quantum multi-key FHE scheme, it is possible to compute a single (multi-key) ciphertext $\hat{\mathsf{ct}}$ that encrypts $C(|\phi_1\rangle, \ldots, |\phi_n\rangle)$ under all public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_n$. Then, if each party additionally drew a key pair $(\mathsf{pk}'_i, \mathsf{sk}'_i)$ for a classical AFS-spooky encryption scheme, and released $\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_n$, where $\tilde{\mathsf{ct}}_i = \mathsf{Enc}(\mathsf{pk}'_i, \mathsf{sk}_i)$ encrypts the i-th party's quantum multi-key FHE secret key under their AFS-spooky encryption public key, then the evaluator can homomorphically evaluate the quantum multi-key FHE decryption circuit (which is classical for classical ciphertexts) with $\hat{\mathsf{ct}}$ hardcoded, where $\hat{\mathsf{ct}}$ is the multi-key ciphertext defined at the beginning of this paragraph. This circuit on input $\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_n$ produces the desired output $\mathsf{ct}'_1, \ldots, \mathsf{ct}'_n$. Finally, note that the classical AFS-spooky encryption scheme must also have distributed setup, and we show (see Section 4.5 in the full version) that one can derive a distributed-setup AFS-spooky encryption scheme from [12] using standard techniques [21].

2.4 Post-Quantum Non-malleable Commitments

In this section, we describe how to obtain constant-round post-quantum non-malleable commitments under the assumption that there exists a natural number $c > 0$ such that quantum polynomial-time adversaries cannot distinguish LWE samples from uniform with advantage better than $\lambda^{-\mathsf{ilog}(c,\lambda)}$, where $\mathsf{ilog}(c, \lambda) = \log\log\cdots_{c \text{ times}}\log(\lambda)$ and λ denotes the security parameter.

We will focus on perfectly binding and computationally hiding constant-round interactive commitments. Loosely speaking, a commitment scheme is said to be non-malleable if no adversary (also called a man-in-the-middle), when participating as a receiver in an execution of an honest commitment $\mathsf{Com}(m)$, can at the same time generate a commitment $\mathsf{Com}(m')$, such that the message m'

is related to the original message m. This is equivalent (assuming the existence of one-way functions with security against quantum adversaries) to a tag-based notion where the commit algorithm obtains as an additional input a tag in $\{0,1\}^\lambda$, and the adversary is restricted to using a tag, or identity, that is different from the tag used to generate its input commitment. We will rely on tag-based definitions throughout this paper. We will also only focus on the *synchronous setting*, where the commitments proceed in rounds, and the man-in-the-middle sends its own message for a specific round before obtaining an honest party's message for the next round.

Before describing our ideas, we briefly discuss existing work on *classically-secure* non-malleable commitments. Unfortunately, existing constructions of constant-round non-malleable commitments against classical adversaries from standard polynomial hardness assumptions [4, 56, 57, 48, 54, 46, 65, 58, 47, 36, 37, 40, 38, 16, 17, 44, 39] either rely on rewinding, or use Barak's non-black-box simulation technique, both of which require the reduction to perform state cloning. As such, known techniques fail to prove quantum security of these constructions.

We now discuss our techniques for constructing post-quantum non-malleable commitments. Just like several classical approaches, we will proceed in two steps.

- We will obtain simple "base" commitment schemes for very small tag/identity spaces from slightly superpolynomial hardness assumptions.
- Then assuming polynomial hardness of LWE against quantum adversaries, and making use of constant-round post-quantum zero-knowledge arguments, we will convert non-malleable commitments for a small tag space into commitments for a larger tag space, while only incurring a constant round overhead.

For the base schemes, there are known classical constructions [58] that assume hardness of LWE against 2^{λ^δ}-size adversaries, where λ denotes the security parameter and $0 < \delta < 1$ is a constant. We observe that these constructions can be proven secure in the quantum setting, resulting in schemes that are suitable for tag spaces of $O(\log\log\lambda)$ tags.

Tag Amplification. Since an MPC protocol could be executed among up to $\mathsf{poly}(\lambda)$ parties where $\mathsf{poly}(\cdot)$ is an arbitrary polynomial, we end up requiring non-malleable commitments suitable for tag spaces of $\mathsf{poly}(\lambda)$. This is obtained by combining classical tools for amplifying tag spaces [22] with constant round post-quantum zero-knowledge protocols. Our tag amplification protocol, on input a scheme with tag space $2t$, outputs a scheme with tag space 2^t, for any $t \leq \mathsf{poly}(\lambda)$. This follows mostly along the lines of existing classical protocols, and as such we do not discuss the protocol in detail here. Our protocol can be found in Section 7.3 in the full version.

Base Schemes from $\lambda^{-\mathsf{ilog}(c,\lambda)}$ Hardness. Returning to the question of constructing appropriate base schemes, we also improve the assumption from 2^{λ^δ}-quantum hardness of LWE (that follows based on [58]) to the mildly superpolynomial hardness assumption discussed at the beginning of this subsection. Recall

that we will only need to assume that there exists an (explicit) natural number $c > 0$ such that quantum polynomial time adversaries cannot distinguish LWE samples from uniform with advantage better than $\mathsf{negl}(\lambda^{\mathsf{ilog}(c,\lambda)})$ where $\mathsf{ilog}(c, \lambda) = \log\log\cdots_{c \text{ times}}\log(\lambda)$. Our base scheme will only be suitable for identities in $\mathsf{ilog}(c+1, \lambda)$, where $c > 0$ is a natural number, independent of λ. We will then repeatedly apply the tag amplification process referred to above to boost the tag space to 2^λ, by adding only a constant number of rounds.

To build our base scheme, we take inspiration from the classically secure non-malleable commitments of Khurana and Sahai [45]. However, beyond considering quantum as opposed to classical adversaries, our protocol and analysis will have the following notable differences from [45]:

- The work of [45] relies on sub-exponential hardness (i.e. 2^{λ^δ} security), which is stronger than the type of superpolynomial hardness we assume. This is primarily because [45] were restricted to two rounds, but we can improve parameters while allowing for a larger constant number of rounds.
- [45] build a reduction that rewinds an adversary to the beginning of the protocol, and executes the adversary several times, repeatedly sampling the adversary's initial state. This may be undesirable in the quantum setting.[6] On the other hand, we have a simpler fully straight-line reduction that only needs to run the adversary once.

Specifically, following [45], we will establish *an erasure channel* between the committer and receiver that transmits the committed message to the receiver with probability ϵ. To ensure that the commitment satisfies hiding, ϵ is chosen to be a value that is negligible in λ. At the same time, the exact value of ϵ is determined by the identity (tag) of the committer. Recall that $\mathsf{tag} \in [1, \mathsf{ilog}(c+1, \lambda)]$. We will set $\epsilon = \eta^{-\mathsf{tag}}$ where $\eta = \lambda^{\mathsf{ilog}(c+1,\lambda)}$ is a superpolynomial function of λ.

Next, for simplicity, we restrict ourselves to a case where the adversary's tag (which we denote by tag') is smaller than that of the honest party (which we denote by tag). In this case, the adversary's committed message is transmitted with probability $\epsilon' = \eta^{-\mathsf{tag}'}$, whereas the honest committer's message is transmitted with probability only $\epsilon = \eta^{-\mathsf{tag}}$, which is smaller than ϵ'.

We set this up so that the transcript of an execution transmits the adversary's message with probability ϵ' (over the randomness of the honest receiver), and on the other hand, an honestly committed message will remain hidden except with probability $\epsilon < \epsilon'$ (over the randomness of the honest committer). This gap in the probability of extraction will help us argue non-malleability, using a proof strategy that bears resemblance to the proof technique in [9] (who relied on stronger assumptions to achieve such a gap in the non-interactive setting).

We point out one subtlety in our proof that does not appear in [9]. We must rule out a man-in-the-middle adversary that on the one hand, does not commit to a related message if its message was successfully transmitted, but on the

[6] In particular this state may not always be efficiently sampleable, in which case it would be difficult to build an efficient reduction.

other hand, can successfully perform a mauling attack if its message was not transmitted. To rule out such an adversary, just like [45], we will design our erasure channel so that the adversary cannot distinguish transcripts where his committed message was transmitted from those where it wasn't.

Finally, our erasure channel can be cryptographically established in a manner similar to prior work [45, 42, 3] via an indistinguishability-based variant of two-party secure function evaluation, that can be based on quantum hardness of LWE. Specifically, we would like to ensure that the SFE error is (significantly) smaller than the transmission probabilities of our erasure channels: therefore, we will set parameters so that SFE error is $\lambda^{-\mathsf{ilog}(c,\lambda)}$. We refer the reader to Section 7 in the full version for additional details about our construction.

On Super-Constant Rounds from Polynomial Hardness. We also observe that for any $t(\lambda) \leq \mathsf{poly}(\lambda)$, non-malleable commitments for tag space of size $t(\lambda)$ can be obtained in $O(t(\lambda))$ rounds based on any extractable commitment using ideas from [22, 15], where only one party speaks in every round. These admit a straight-line reduction, and can be observed to be quantum-secure. As such, based on quantum polynomial hardness of LWE and quantum FHE, we can obtain a base protocol for $O(\log\log\ldots_{\text{c times}}\log\lambda)$ tags requiring $O(\log\log\ldots_{\text{c times}}\log\lambda)$ rounds, for any constant $c \in \mathbb{N}$. Applying our tag-amplification compiler to this base protocol makes it possible to increase the tag space to 2^λ while only adding a constant number of rounds. Therefore, this technique gives $O(\log\log\ldots_{\text{c times}}\log\lambda)$ round non-malleable commitments for exponentially large tags from quantum polynomial hardness. It also yields constant round non-malleable commitments for a constant number of tags from polynomial hardness.

2.5 Putting Things Together

Finally, we show how to combine the primitives described above to obtain a constant-round coin-flipping protocol that supports straight-line simulation. As we saw above, in the setting of multi-verifier zero-knowledge, simultaneously simulating the view of multiple parties without rewinding can be quite challenging, so a careful protocol and proof is needed.

Recall the outline presented at the beginning of this section, where each party first commits to a uniformly random string, then broadcasts the committed message, and finally proves in ZK that the message broadcasted is equal to the previously committed message. If all proofs verify, then the common output is the XOR of all broadcasted strings. Recall also that the coin-tossing protocol should be *fully-simulatable*. This means that a simulator should be able to force the common output to be a particular uniformly drawn string given to it as input.

It turns out that in order to somehow force a particular output, the simulator should be able to *simultaneously extract in advance* all the messages that adversarial parties committed to. In particular, we require commitments where a simulator can extract from multiple committers committing in parallel. Here,

we will rely on our parallel extractable commitment described above. Note that we will also need to simulate the subsequent zero-knowledge arguments given by the malicious parties in parallel, and thus we instantiate these with our parallel zero-knowledge argument described above. However, an issue remains. What if an adversary could somehow *maul* an honest party's commitment to a related message and then broadcast that commitment as their own? This could bias the final outcome away from uniformly random.

Thus, we need to introduce some form of non-malleability into the protocol. Indeed, we will add another step at the beginning where each party commits to its message c_i and some randomness r_i using our post-quantum many-to-one non-malleable commitment.[7] Each party will then commit to c_i again with our extractable commitment, using randomness r_i. Finally, each party proves in zero-knowledge that the previous commitments were consistent.

This protocol can be proven to be fully simulatable. Intuitively, even though the simulator changes the behavior of honest players in order to extract from the adversary's commitments and then later force the appropriate output, the initial non-malleable commitments given by the adversary must not change in a meaningful way, due the the guarantee of non-malleability. However, additional subtleties arise in the proof of security. In particular, during the hybrids the simulator will first have to simulate the honest party zero-knowledge arguments, before changing the honest party commitments in earlier stages. However, when changing an honest party's commitment, we need to rely on non-malleability to ensure that the malicious party commitments will not also change in a non-trivial way. Here, we use a proof technique that essentially invokes soundness of the adversary's zero-knowledge arguments at an earlier hybrid but allows us to nevertheless rely on non-malleable commitments to enforce that the adversary behaves consistently in all future hybrids. More discussion and a formal analysis can be found in Section 8 in the full version.

2.6 Related Work

Classical secure multi-party computation was introduced and shown to be achievable in the two-party setting by [67] and in the multi-party setting by [33]. Since these seminal works, there has been considerable interest in reducing the round complexity of classical protocols. In the setting of malicious security against a dishonest majority, [49] gave the first *constant*-round protocol for two-party computation, and [43] gave the first constant-round protocol for multi-party computation. Since then, there has been a long line of work improving on the exact round complexity and assumptions necessary for classical multi-party computation (see e.g. [55, 27]).

Post-quantum classical protocols. The above works generally focus on security against *classical* polynomial-time adversaries. Another line of work, most relevant to the present work, has considered the more general goal of proving

[7] Above we described a construction of one-to-one non-malleable commitment, though a hybrid argument [48] shows that one-to-one implies many-to-one.

the security of classical protocols against arbitrary *quantum* polynomial-time adversaries.

This study was initiated by van de Graaf [63], who observed that the useful rewinding technique often used to prove zero-knowledge in the classical setting may be problematic in the quantum setting. In a breakthrough work, Watrous [64] showed that several well-known classical zero-knowledge protocols are in fact zero-knowledge against quantum verifiers, via a careful rewinding argument. However, these protocols require a polynomial number of rounds to achieve negligible security against quantum attackers. Later, Unruh [62] developed a more powerful rewinding technique that suffices to construct classical zero-knowledge *proofs of knowledge* secure against quantum adversaries, though still in a polynomial number of rounds. In a recent work, [10] managed to construct a constant-round post-quantum zero-knowledge protocol, under assumptions similar to those required to obtain classical fully-homomorphic encryption. In another recent work, [1] constructed a constant-round protocol that is zero-knowledge against quantum verifiers under the quantum LWE assumption, though soundness holds against only classical provers.

There has also been some work on the more general question of post-quantum secure computation. In particular, [20] used the techniques developed in [64] to build a two-party coin-flipping protocol, and [51, 41] constructed general two-party computation secure against quantum adversaries, in a polynomial number of rounds. More recently, [10] gave a *constant*-round two-party coin-flipping protocol, with full simulation of one party. However, prior to this work, nothing was known in the most general setting of post-quantum multi-party computation (in the plain model).

Finally, we remark that post-quantum classical protocols do exist in the literature, as long as some form of trusted setup is available. For example, the two-round protocol of [53] from LWE is in the programmable common random string model, and was shown to be semi-maliciously secure via straight-line simulation. Thus, applying the semi-malicious to malicious compiler of [2] instantiated with a NIZK from LWE [60] gives a post-quantum maliciously secure protocol in the common random string model from the quantum hardness of LWE. Another example is the maliciously secure OT-based two-round protocol of [28, 7] instantiated with maliciously-secure oblivious transfer from LWE [61].

Quantum protocols. Yet another line of work focuses on protocols for securely computing *quantum* circuits. General multi-party quantum computation was shown to be achievable in the information-theoretic setting (with honest majority) in the works of [19, 5]. In the computational setting, [25] gave a two-party protocol secure against a quantum analogue of semi-honest adversaries, and [26] extended security of two-party quantum computation to the malicious setting. In a recent work [23] constructed a maliciously secure multi-party protocol for computing quantum circuits, assuming the existence of a maliciously secure post-quantum classical MPC protocol. We remark that all of the above protocols operate in a polynomial number of rounds.

3 Quantum-Secure Multi-Committer Extractable Commitment

In this section, we follow the outline presented in Section 2.2 to construct a commitment scheme that allows for simultaneous extraction from multiple parallel committers. The protocol is somewhat more involved than the high-level description given earlier, so we briefly highlight the differences.

First, the committer is instructed to (non-interactively) commit to its message and trapdoor at the very beginning of the protocol. We use these commitments to take advantage of non-uniformity in the reductions between hybrids in the extractability proof. In particular, hybrids that come before the step where the simulator goes "under the hood" of the FHE may still need access to the trapdoor and commitment, and this can be given to any reduction via non-uniform advice consisting of each committer's first message and corresponding openings.

Next, the CDS described earlier is replaced with a function-hiding secure function evaluation (SFE) protocol. In order to rule out the malleability attack mentioned in Section 2.2, where a malicious receiver mauls the AFS-spooky encryption of the committer's trapdoor into an SFE encryption of the trapdoor, we do the following. The first message sent by the receiver to each committer C_i will actually be a commitment to some key k_i of a generic secret-key encryption scheme. After C_i sends its AFS-spooky encryption ciphertext and compute and compare obfuscation, the receiver prepares and sends a secret-key encryption of an arbitrary message. Then, the receiver's input to the SFE consists of the opening to its earlier commitment k_i, and the SFE checks if the secret-key encryption sent by the receiver is actually an encryption of the committer's trapdoor under secret key k_i. If so, it returns the lock and otherwise it returns $\perp$. This setup ensures that a malicious receiver cannot maul the AFS-spooky encryption of the committer's trapdoor, for the following reason. If it could, then a non-uniform reduction to the semantic security of AFS-spooky encryption may obtain the receiver's committed k_i as advice and decrypt the receiver's secret-key encryption to obtain the trapdoor. Of course, this assumes the receiver actually acted explainably in sending a valid commitment at the beginning of the protocol, and this is ensured by the opening check performed under the SFE. We note that this mechanism is somewhat different than what was presented in [10], as they directly build a zero-knowledge argument (i.e. without first constructing a stand-alone extractable commitment) and are able to take advantage of witness indistinguishability to enforce explainable behavior.

Compliant Distinguishers. Finally, we discuss the issue of *committer* explainability. Recall from the high-level overview that a simulator is able to extract from a committer by homomorphically evaluating its code on an AFS-spooky encryption ciphertext *generated by the committer*. Thus, if the committer acts arbitrarily maliciously and does not return a well-formed ciphertext, the extraction may completely fail. Again, [10] address this issue by only analyzing their commitment within the context of a larger zero-knowledge argument protocol,

and having the verifier prove to the prover using a witness indistinguishable proof that it performed the commitment explainably.

Thus, without adding zero-knowledge and performing [32]-style analysis to handle non-explainable and aborting committers, we will only obtain extractability against explainable committers. However, since we will be using this protocol inside larger protocols where participants are not assumed to be acting explainably, restricting the class of committers we consider in our definition is problematic. We instead consider arbitrary committers but restrict the class of *distinguishers* (who are supposed to decide whether they received the view of a committer interacting in the real protocol or the view of a committer interacting with the extractor) to those that always output 0 on input a non-explainable transcript. In other words, any advantage these distinguishers may have must be coming from their behavior on input explainable views. Even though checking whether a particular view is explainable or not is not efficient, it turns out that this definition lends itself quite nicely to composition, since one can use witness indistinguishability/zero-knowledge to construct provably compliant distinguishers between hybrids for the larger protocols.

For completeness, and because post-quantum multi-committer extractable commitments may be of independent interest, we also show in Appendix D in the full version how to add zero-knowledge within the extractable commitment protocol itself to obtain security against arbitrary committers.

3.1 Definition

Definition 1 (Quantum-Secure Multi-Committer Extractable Commitment). *A quantum-secure multi-committer extractable commitment scheme is a pair* (C, R) *of classical PPT interactive Turing machines. In the commit phase,* R *interacts with* n *copies* $\{\mathsf{C}_i\}_{i\in[n]}$ *of* C *(who do not interact with each other) on common input* 1^λ *and* 1^n*, with each* C_i *additionally taking a private input* $m_i \in \{0,1\}^*$*. This produces a transcript* τ*, which may be parsed as a set of* n *transcripts* $\{\tau_i\}_{i\in[n]}$*, one for each set of messages exchanged between* R *and* C_i*. In the decommitment phase, each* C_i *outputs* m_i *along with its random coins* r_i*, and* R *on input* $(1^\lambda, \tau_i, m_i, r_i)$ *either accepts or rejects. The scheme should satisfy the following properties.*

- **Perfect Correctness:** *For any* $\lambda, n \in \mathbb{N}, i \in [n]$*,*

  $$\Pr[\mathsf{R}(1^\lambda, \tau_i, m_i, r_i) = 1 \mid \{\tau_i\}_{i\in[n]} \leftarrow \langle \mathsf{R}, \mathsf{C}_1(m_1; r_1), \ldots, \mathsf{C}_n(m_n; r_n)\rangle(1^\lambda, 1^n)] = 1.$$

- **Perfect Binding:** *For any* $\lambda \in \mathbb{N}$ *and string* $\tau \in \{0,1\}^*$*, there does not exist* (m, r) *and* (m', r') *with* $m \neq m'$ *such that* $\mathsf{R}(1^\lambda, \tau, m, r) = \mathsf{R}(1^\lambda, \tau, m', r') = 1$*.*
- **Quantum Computational Hiding:** *For any non-uniform quantum polynomial-size receiver* $\mathsf{R}^* = \{\mathsf{R}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$*, any polynomial* $\ell(\cdot)$*, and any sequence of sets of strings* $\{m^{(0)}_{\lambda,1}, \ldots, m^{(0)}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$*,* $\{m^{(1)}_{\lambda,1}, \ldots, m^{(1)}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$ *where each* $|m^{(b)}_{\lambda,i}| =$

$\ell(\lambda)$,

$$\{\mathsf{VIEW}_{\mathsf{R}^*_\lambda}(\langle \mathsf{R}^*_\lambda(\rho_\lambda), \mathsf{C}_1(m^{(0)}_{\lambda,1}), \ldots, \mathsf{C}_n(m^{(0)}_{\lambda,n})\rangle(1^\lambda, 1^n))\}_{\lambda,n\in\mathbb{N}}$$
$$\approx_c \{\mathsf{VIEW}_{\mathsf{R}^*_\lambda}(\langle \mathsf{R}^*_\lambda(\rho_\lambda), \mathsf{C}_1(m^{(1)}_{\lambda,1}), \ldots, \mathsf{C}_n(m^{(1)}_{\lambda,n})\rangle(1^\lambda, 1^n))\}_{\lambda,n\in\mathbb{N}}.$$

The extractability property will require the following two definitions. First, for any adversary $\mathsf{C}^* = \{\mathsf{C}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *representing a subset* $I \subseteq [n]$ *of* n *committers, any honest party messages* $\{m_i\}_{i\notin I}$*, and any security parameter* $\lambda \in \mathbb{N}$*, define* $\mathsf{VIEW}^{\mathsf{msg}}_{\mathsf{C}^*_\lambda}(\langle \mathsf{R}, \mathsf{C}^*_\lambda(\rho_\lambda), \{\mathsf{C}_i(m_i)\}_{i\notin I}\rangle(1^\lambda, 1^n))$ *to consist of the following.*

1. *The view of* C^*_λ *on interaction with the honest receiver* R *and set* $\{\mathsf{C}_i(m_i)\}_{i\notin I}$ *of honest parties; this view includes a set of transcripts* $\{\tau_i\}_{i\in I}$ *and a state* st*.*
2. *A set of strings* $\{m_i\}_{i\in I}$*, where each* m_i *is defined relative to* τ_i *as follows. If there exists* m'_i, r_i *such that* $\mathsf{R}(1^\lambda, \tau_i, m'_i, r_i) = 1$*, then* $m_i = m'_i$*, otherwise,* $m_i = \bot$*.*

Next, we consider distinguishers $\mathsf{D} = \{\mathsf{D}_\lambda, \sigma_\lambda\}_{\lambda\in\mathbb{N}}$ *that take as input a sample* $(\{\tau_i\}_{i\in I}, \mathsf{st}, \{m_i\}_{i\in I})$ *from the distribution just described. We say that* D *is* compliant *if whenever* $\{\tau_i\}_{i\in I}$ *is not an explainable transcript with respect to the set* I*,* D *outputs 0 with overwhelming probability (over the randomness of* D*).*

– **Multi-Committer Extractability:** *There exists a quantum expected-polynomial-time extractor* Ext *such that for any* compliant *non-uniform polynomial-size quantum distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda, \sigma_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $\mu(\cdot)$*, such that for all adversaries* $\mathsf{C}^* = \{\mathsf{C}^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *representing a subset of* n *committers, namely,* $\{\mathsf{C}_i\}_{i\in I}$ *for some set* $I \subseteq [n]$*, the following holds for all polynomial-size sequences of inputs* $\{\{m_{i,\lambda}\}_{i\notin I}\}_{\lambda\in\mathbb{N}}$ *and* $\lambda \in \mathbb{N}$*.*

$$\big|\Pr[\mathsf{D}_\lambda(\mathsf{VIEW}^{\mathsf{msg}}_{\mathsf{C}^*_\lambda}(\langle \mathsf{R}, \mathsf{C}^*_\lambda(\rho_\lambda), \{\mathsf{C}_i(m_{i,\lambda})\}_{i\notin I}\rangle(1^\lambda, 1^n)), \sigma_\lambda) = 1]$$
$$- \Pr[\mathsf{D}_\lambda(\mathsf{Ext}(1^\lambda, 1^n, I, \mathsf{C}^*_\lambda, \rho_\lambda), \sigma_\lambda) = 1]\big| \leq \mu(\lambda).$$

Remark 1. Observe that the above definition of quantum computational hiding does not consider potentially malicious committers that interact in the protocol to try to gain information about commitments made by other committers. This is without loss of generality, since all communication occurs between R and some C_i. In particular, no messages are sent between any C_i and C_j.

3.2 Construction

Ingredients: All of the following are assumed to be quantum-secure, and the construction is presented in Protocol 2.

– A non-interactive perfectly-binding commitment Com.
– A secret-key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$.[8]

[8] We use the syntax that for key k, a ciphertext of message m is computed as $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$ and decrypted as $m := \mathsf{Dec}(k, \mathsf{ct})$.

– A compute-and-compare obfuscator Obf.
– A quantum AFS-spooky encryption scheme with distributed setup ($\mathsf{Spooky.Setup}$,$\mathsf{Spooky.KeyGen}$, $\mathsf{Spooky.Enc}$, $\mathsf{Spooky.QEnc}$, $\mathsf{Spooky.Eval}$, $\mathsf{Spooky.Dec}$, $\mathsf{Spooky.QDec}$).
– A two-message function-hiding secure function evaluation scheme ($\mathsf{SFE.Gen}, \mathsf{SFE.Enc}, \mathsf{SFE.Eval}, \mathsf{SFE.Dec}$).

Protocol 2

Common input: $1^\lambda, 1^n$.
C_i**'s additional input:** A string m_i.

1. Each C_i computes $\mathsf{td}_i \leftarrow U_\lambda$ and sends $\mathsf{c}_i^{(\mathsf{msg})} \leftarrow \mathsf{Com}(1^\lambda, m_i)$, $\mathsf{c}_i^{(\mathsf{td})} \leftarrow \mathsf{Com}(1^\lambda, \mathsf{td}_i)$ to R.
2. For each $i \in [n]$, R computes $k_i, r_i \leftarrow U_\lambda$ and sends $\mathsf{c}_i^{(\mathsf{key})} := \mathsf{Com}(1^\lambda, k_i; r_i)$ to C_i.
3. Each C_i computes and sends $\mathsf{pp}_i \leftarrow \mathsf{Spooky.Setup}(1^\lambda)$ to R.
4. R defines $\mathsf{pp} := \{\mathsf{pp}_i\}_{i\in[n]}$, and sends pp to each C_i. Each C_i checks that the pp_i it received matches the pp_i it sent in Step 3, and if not, it aborts.
5. Each C_i computes
   – $\mathsf{lk}_i \leftarrow U_\lambda$,
   – $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Spooky.KeyGen}(1^\lambda, \mathsf{pp})$,
   – $\mathsf{ct}_i \leftarrow \mathsf{Spooky.Enc}(\mathsf{pk}_i, \mathsf{td}_i)$,
   – and $\widetilde{\mathbf{CC}}_i \leftarrow \mathsf{Obf}\left(\mathbf{CC}[\mathsf{Spooky.Dec}(\mathsf{sk}_i, \cdot), \mathsf{lk}_i, (\mathsf{sk}_i, m_i)]\right)$,

   and sends $(\mathsf{pk}_i, \mathsf{ct}_i, \widetilde{\mathbf{CC}}_i)$ to R.
6. For each $i \in [n]$, R computes $\mathsf{ct}_i^{(\mathsf{td})} \leftarrow \mathsf{Enc}(k_i, 0^\lambda)$, $\mathsf{dk}_i \leftarrow \mathsf{SFE.Gen}(1^\lambda)$, and $\mathsf{ct}_i^{(\mathsf{SFE})} \leftarrow \mathsf{SFE.Enc}(\mathsf{dk}_i, (k_i, r_i))$ and sends $(\mathsf{ct}_i^{(\mathsf{td})}, \mathsf{ct}_i^{(\mathsf{SFE})})$ to C_i.
7. Define the circuit $\mathsf{C}[\mathsf{c}_i^{(\mathsf{key})}, \mathsf{ct}_i^{(\mathsf{td})}, \mathsf{td}_i, \mathsf{lk}_i](\cdot)$ to take as input (k_i, r_i), check if $\mathsf{c}_i^{(\mathsf{key})}$ opens to k_i with opening r_i and if $\mathsf{td}_i = \mathsf{Dec}(k_i, \mathsf{c}_i^{(\mathsf{td})})$, and if so output lk_i, and otherwise output $\perp$. Each C_i computes and sends $\widehat{\mathsf{ct}}_i^{(\mathsf{SFE})} \leftarrow \mathsf{SFE.Eval}(\mathsf{C}[\mathsf{c}_i^{(\mathsf{key})}, \mathsf{ct}_i^{(\mathsf{td})}, \mathsf{td}_i, \mathsf{lk}_i], \mathsf{ct}_i^{(\mathsf{SFE})})$.

Fig. 2: Constant round post-quantum multi-committer extractable commitment.

Analysis. We state the security of our scheme in the following and we refer the reader to the full version of this work for the proofs.

Lemma 1. *Protocol 2 is quantum computational hiding.*

Lemma 2. *Protocol 2 is multi-committer extractable.*

References

1. Ananth, P., Placa, R.L.L.: Secure quantum extraction protocols. Cryptology ePrint Archive, Report 2019/1323 (2019), `https://eprint.iacr.org/2019/1323`

2. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4$_2$9
3. Badrinarayanan, S., Fernando, R., Jain, A., Khurana, D., Sahai, A.: Statistical ZAP arguments. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 642–667. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_22, https://doi.org/10.1007/978-3-030-45727-3_22
4. Barak, B.: Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. In: FOCS 2002. pp. 345–355 (2002)
5. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multiparty quantum computation with (only) a strict honest majority. In: 47th FOCS. pp. 249–260. IEEE Computer Society Press, Berkeley, CA, USA (Oct 21–24, 2006). https://doi.org/10.1109/FOCS.2006.68
6. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: 20th ACM STOC. pp. 1–10. ACM Press, Chicago, IL, USA (May 2–4, 1988). https://doi.org/10.1145/62212.62213
7. Benhamouda, F., Lin, H.: k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 500–532. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018). https://doi.org/10.1007/978-3-319-78375-8$_1$7
8. Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019. pp. 1091–1102 (2019). https://doi.org/10.1145/3313276.3316382, https://doi.org/10.1145/3313276.3316382
9. Bitansky, N., Lin, H.: One-message zero knowledge and non-malleable commitments. In: Theory of Cryptography Conference, TCC 2018, Goa, India, November 11-14, 2018, Proceedings (2018)
10. Bitansky, N., Shmueli, O.: Post-quantum zero knowledge in constant rounds. STOC (2020)
11. Brakerski, Z.: Quantum FHE (almost) as secure as classical. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 67–95. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96878-0$_3$
12. Brakerski, Z., Halevi, S., Polychroniadou, A.: Four round secure computation without setup. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 645–677. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017). https://doi.org/10.1007/978-3-319-70500-2$_2$2
13. Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 609–629. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015). https://doi.org/10.1007/978-3-662-48000-7$_3$0

14. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (abstract) (informal contribution). In: Pomerance, C. (ed.) CRYPTO'87. LNCS, vol. 293, p. 462. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 1988). https://doi.org/10.1007/3-540-48184-2$_4$3
15. Chor, B., Rabin, M.: Achieving independence in logarithmic number of rounds. pp. 260–268 (01 1987). https://doi.org/10.1145/41840.41862
16. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9816, pp. 270–299. Springer (2016). https://doi.org/10.1007/978-3-662-53015-3$_1$0, https://doi.org/10.1007/978-3-662-53015-3_10
17. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Annual International Cryptology Conference. pp. 127–157. Springer (2017)
18. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 630–656. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015). https://doi.org/10.1007/978-3-662-48000-7$_3$1
19. Crépeau, C., Gottesman, D., Smith, A.: Secure multi-party quantum computation. In: 34th ACM STOC. pp. 643–652. ACM Press, Montréal, Québec, Canada (May 19–21, 2002). https://doi.org/10.1145/509907.510000
20. Damgård, I., Lunemann, C.: Quantum-secure coin-flipping and applications. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 52–69. Springer, Heidelberg, Germany, Tokyo, Japan (Dec 6–10, 2009). https://doi.org/10.1007/978-3-642-10366-7$_4$
21. Dodis, Y., Halevi, S., Rothblum, R.D., Wichs, D.: Spooky encryption and its applications. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 93–122. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53015-3$_4$
22. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography (Extended Abstract). In: STOC 1991 (1991)
23. Dulek, Y., Grilo, A.B., Jeffery, S., Majenz, C., Schaffner, C.: Secure multi-party quantum computation with a dishonest majority. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 729–758. Springer (2020). https://doi.org/10.1007/978-3-030-45727-3_25, https://doi.org/10.1007/978-3-030-45727-3_25
24. Dulek, Y., Schaffner, C., Speelman, F.: Quantum homomorphic encryption for polynomial-sized circuits. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 3–32. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53015-3$_1$
25. Dupuis, F., Nielsen, J.B., Salvail, L.: Secure two-party quantum evaluation of unitaries against specious adversaries. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 685–706. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 2010). https://doi.org/10.1007/978-3-642-14623-7$_3$7
26. Dupuis, F., Nielsen, J.B., Salvail, L.: Actively secure two-party evaluation of any quantum operation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS,

vol. 7417, pp. 794–811. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012). https://doi.org/10.1007/978-3-642-32009-5$_4$6

27. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 448–476. Springer, Heidelberg, Germany, Vienna, Austria (May 8–12, 2016). https://doi.org/10.1007/978-3-662-49896-5$_1$6
28. Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 468–499. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018). https://doi.org/10.1007/978-3-319-78375-8$_1$6
29. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press, Bethesda, MD, USA (May 31 – Jun 2, 2009). https://doi.org/10.1145/1536414.1536440
30. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press, Victoria, BC, Canada (May 17–20, 2008). https://doi.org/10.1145/1374376.1374407
31. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013). https://doi.org/10.1007/978-3-642-40041-4$_5$
32. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology **9**(3), 167–190 (Jun 1996)
33. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press, New York City, NY, USA (May 25–27, 1987). https://doi.org/10.1145/28395.28420
34. Goyal, R.: Quantum multi-key homomorphic encryption for polynomial-sized circuits. Cryptology ePrint Archive, Report 2018/443 (2018), `https://eprint.iacr.org/2018/443`
35. Goyal, R., Koppula, V., Waters, B.: Lockable obfuscation. In: Umans, C. (ed.) 58th FOCS. pp. 612–621. IEEE Computer Society Press, Berkeley, CA, USA (Oct 15–17, 2017). https://doi.org/10.1109/FOCS.2017.62
36. Goyal, V.: Constant round non-malleable protocols using one way functions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 695–704. ACM Press, San Jose, CA, USA (Jun 6–8, 2011). https://doi.org/10.1145/1993636.1993729
37. Goyal, V., Lee, C.K., Ostrovsky, R., Visconti, I.: Constructing non-malleable commitments: A black-box approach. In: FOCS (2012)
38. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: STOC. pp. 1128–1141. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2897518.2897657, `http://doi.acm.org/10.1145/2897518.2897657`
39. Goyal, V., Richelson, S.: Non-malleable commitments using goldreich-levin list decoding. In: Zuckerman, D. (ed.) 60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019. pp. 686–699. IEEE Computer Society (2019). https://doi.org/10.1109/FOCS.2019.00047, `https://doi.org/10.1109/FOCS.2019.00047`

40. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: FOCS 2014. pp. 41–50 (2014). https://doi.org/10.1109/FOCS.2014.13
41. Hallgren, S., Smith, A., Song, F.: Classical cryptographic protocols in a quantum world. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 411–428. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2011). https://doi.org/10.1007/978-3-642-22792-$9_2$3
42. Kalai, Y.T., Khurana, D., Sahai, A.: Statistical witness indistinguishability (and more) in two messages. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 34–65. Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_2, https://doi.org/10.1007/978-3-319-78372-7_2
43. Katz, J., Ostrovsky, R., Smith, A.: Round efficiency of multi-party computation with a dishonest majority. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 578–595. Springer, Heidelberg, Germany, Warsaw, Poland (May 4–8, 2003). https://doi.org/10.1007/3-540-39200-$9_3$6
44. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10678, pp. 139–171. Springer (2017). https://doi.org/10.1007/978-3-319-70503-3_5, https://doi.org/10.1007/978-3-319-70503-3_5
45. Khurana, D., Sahai, A.: How to achieve non-malleability in one or two rounds. In: Umans, C. (ed.) 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017. pp. 564–575. IEEE Computer Society (2017). https://doi.org/10.1109/FOCS.2017.58, https://doi.org/10.1109/FOCS.2017.58
46. Lin, H., Pass, R.: Non-malleability Amplification. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. pp. 189–198. STOC '09 (2009)
47. Lin, H., Pass, R.: Constant-round non-malleable commitments from any one-way function. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 705–714. ACM Press, San Jose, CA, USA (Jun 6–8, 2011). https://doi.org/10.1145/1993636.1993730
48. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent Non-malleable Commitments from Any One-Way Function. In: TCC 2008. pp. 571–588
49. Lindell, Y.: Parallel coin-tossing and constant-round secure two-party computation. Journal of Cryptology **16**(3), 143–184 (Jun 2003). https://doi.org/10.1007/s00145-002-0143-7
50. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Karloff, H.J., Pitassi, T. (eds.) 44th ACM STOC. pp. 1219–1234. ACM Press, New York, NY, USA (May 19–22, 2012). https://doi.org/10.1145/2213977.2214086
51. Lunemann, C., Nielsen, J.B.: Fully simulatable quantum-secure coin-flipping and applications. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 11. LNCS, vol. 6737, pp. 21–40. Springer, Heidelberg, Germany, Dakar, Senegal (Jul 5–7, 2011)
52. Mahadev, U.: Classical homomorphic encryption for quantum circuits. In: Thorup, M. (ed.) 59th FOCS. pp. 332–338. IEEE Computer Society Press, Paris, France (Oct 7–9, 2018). https://doi.org/10.1109/FOCS.2018.00039

53. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 735–763. Springer, Heidelberg, Germany, Vienna, Austria (May 8–12, 2016). https://doi.org/10.1007/978-3-662-49896-5$_2$6
54. Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive One-Way Functions and Applications. In: Advances in Cryptology — CRYPTO '08. pp. 57–74 (2008)
55. Pass, R.: Bounded-concurrent secure multi-party computation with a dishonest majority. In: Babai, L. (ed.) 36th ACM STOC. pp. 232–241. ACM Press, Chicago, IL, USA (Jun 13–16, 2004). https://doi.org/10.1145/1007352.1007393
56. Pass, R., Rosen, A.: Concurrent Non-Malleable Commitments. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of ComputerScience. pp. 563–572. FOCS '05 (2005)
57. Pass, R., Rosen, A.: New and Improved Constructions of Nonmalleable Cryptographic Protocols. SIAM J. Comput. **38**(2), 702–752 (2008)
58. Pass, R., Wee, H.: Constant-round non-malleable commitments from sub-exponential one-way functions. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 638–655. Springer, Heidelberg, Germany, French Riviera (May 30 – Jun 3, 2010). https://doi.org/10.1007/978-3-642-13190-5$_3$2
59. Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 217–238. Springer, Heidelberg, Germany, Beijing, China (Oct 31 – Nov 3, 2016). https://doi.org/10.1007/978-3-662-53644-5$_9$
60. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2019, Part I. pp. 89–114. LNCS, Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019). https://doi.org/10.1007/978-3-030-26948-7$_4$
61. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2008). https://doi.org/10.1007/978-3-540-85174-5$_3$1
62. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4$_1$0
63. Van De Graaf, J.: Towards a Formal Definition of Security for Quantum Protocols. Ph.D. thesis, CAN (1998), aAINQ35648
64. Watrous, J.: Zero-knowledge against quantum attacks. SIAM J. Comput. **39**(1), 25–58 (May 2009). https://doi.org/10.1137/060670997, `https://doi.org/10.1137/060670997`
65. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: 51st FOCS. pp. 531–540. IEEE Computer Society Press, Las Vegas, NV, USA (Oct 23–26, 2010). https://doi.org/10.1109/FOCS.2010.87
66. Wichs, D., Zirdelis, G.: Obfuscating compute-and-compare programs under LWE. In: Umans, C. (ed.) 58th FOCS. pp. 600–611. IEEE Computer Society Press, Berkeley, CA, USA (Oct 15–17, 2017). https://doi.org/10.1109/FOCS.2017.61
67. Yao, A.C.C.: Protocols for secure computations (extended abstract). In: 23rd FOCS. pp. 160–164. IEEE Computer Society Press, Chicago, Illinois (Nov 3–5, 1982). https://doi.org/10.1109/SFCS.1982.38
68. Yao, A.C.C.: How to generate and exchange secrets. In: FOCS (1986)