One-Way Functions Imply Secure Computation in a Quantum World

James Bartusek⋆, Andrea Coladangelo⋆⋆, Dakshita Khurana⋆ ⋆ ⋆, and Fermi Ma†

Abstract. We prove that quantum-hard one-way functions imply *simulation-secure* quantum oblivious transfer (QOT), which is known to suffice for secure computation of arbitrary quantum functionalities. Furthermore, our construction only makes *black-box* use of the quantum-hard one-way function.

Our primary technical contribution is a construction of *extractable and equivocal* quantum bit commitments based on the black-box use of quantum-hard one-way functions in the standard model. Instantiating the Crépeau-Kilian (FOCS 1988) framework with these commitments yields simulation-secure QOT.

1 Introduction

The complexity of cryptographic primitives is central to the study of cryptography. Much of the work in the field focuses on establishing *reductions* between different primitives, typically building more sophisticated primitives from simpler ones. Reductions imply relative measures of complexity among different functionalities, and over the years have resulted in an expansive hierarchy of assumptions and primitives, as well as separations between them.

One-way functions (OWFs) lie at the center of cryptographic complexity: their existence is the minimal assumption necessary for nearly all classical cryptography [28, 23, 22]. One-way functions are equivalent to so-called "minicrypt" primitives like pseudorandom generators, pseudorandom functions and symmetric encryption; but provably cannot imply key exchange when used in a black-box way [24, 3]. Thus, the existence of key exchange is believed to be a stronger assumption than the existence of one-way functions. Oblivious transfer (OT) is believed to be *even stronger*: it implies key exchange, but cannot be obtained from black-box use of a key exchange protocol [29].

The importance of OT stems from the fact that it can be used to achieve secure computation, which is a central cryptographic primitive with widespread applications. In a nutshell, secure computation allows mutually distrusting participants to compute any public function over their joint private inputs while revealing no private information beyond the output of the computation.

⋆ UC Berkeley. Email: bartusek.james@gmail.com
⋆⋆ UC Berkeley. Email: andrea.coladangelo@gmail.com
⋆ ⋆ ⋆ UIUC. Email: dakshita@illinois.edu
† Princeton University and NTT Research. Email: fermima@alum.mit.edu

The Quantum Landscape. The landscape of cryptographic possibilities changes significantly when participants have quantum computation and communication capabilities. For one, *unconditionally* secure key distribution — commonly known as *quantum key distribution* (QKD) — becomes possible [5]. Moreover, *quantum* oblivious transfer (QOT) is known to be achievable from special types of commitments, as we discuss next.

Crépeau and Kilian [11] first proposed a protocol for QOT using quantum bit commitments. The central idea in these QKD and QOT protocols is the use of (what are now known as) "BB84 states". These are single qubit states encoding either 0 or 1 in either the computational or Hadamard basis. Crucially, measuring (or essentially attempting to copy the encoded bit) in the wrong basis completely destroys information about the encoded bit. Then [6] presented a transmission-error resistant version of the [11] protocol. These protocols did not come with a proof of security, but subsequently Mayers and Salvail [31] proved that the [11] protocol is secure against a restricted class of attackers that only perform single-qubit measurements. This was later improved by Yao [37], who extended the [31] result to handle general quantum adversaries.

By an unfortunate historical accident, the aforementioned security proofs claimed the [11] QOT could be *information-theoretically secure*, since at the time it was believed that information-theoretic quantum bit commitment was possible [9]. Several years later, Mayers [30] and Lo and Chau [27] independently proved the impossibility of information-theoretic quantum bit commitment, and as a consequence, the precise security of [11] QOT was once again unclear. This state of affairs remained largely unchanged until 2009, when Damgard, Fehr, Lunemann, Salvail, and Schaffner [13] proved that bit commitment schemes satisfying certain additional properties, namely *extraction and equivocation*, suffice to instantiate [11] QOT. [13] called their commitments *dual-mode* commitments, and provided a construction based on the quantum hardness of the learning with errors (QLWE) assumption. We remark that assumptions about the hardness of specific problems like QLWE are qualitatively even worse than general assumptions like QOWFs and QOT. Thus, the following basic question remains:

Do quantum-hard one-way functions suffice for quantum oblivious transfer?

Quantum OT: The Basis of Secure Quantum Computation. There is a natural extension of secure computation to the quantum world, where Alice and Bob wish to compute a *quantum* circuit on (possibly entangled) *quantum* input states. This setting, usually referred to as secure *quantum* computation, has been previously studied and in fact has a strong tradition in the quantum cryptography literature.

[10, 4] constructed unconditional maliciously-secure *multi-party* quantum computation with honest majority. The setting where half (or more) of the players are malicious requires computational assumptions due to the impossibility of unconditionally secure quantum bit commitment [30, 27].

In this computational setting, [16, 17] showed the feasibility of two-party quantum computation (2PQC) assuming post-quantum OT. More recently, [15] constructed maliciously-secure general multi-party quantum computation (MPQC) secure against a *dishonest* majority from any maliciously-secure post-quantum multi-party computation (MPC) protocol for classical functionalities, which can itself be obtained from post-quantum OT [2].

Nevertheless, the following natural question has remained unanswered:

Can secure (quantum) computation be obtained from quantum-hard one-way functions?

1.1 Our Results

Our main result is the following:

Quantum oblivious transfer can be based on the assumption that quantum-hard one-way functions exist.

In fact, we prove a stronger result: we show that quantum oblivious transfer can be based on the *black-box use of any statistically binding, quantum computationally hiding commitment*. Such commitments can be based on the black-box use of quantum-hard one-way functions. This in turn implies secure two-party computation of classical functionalities, in the presence of quantum computation and communication capabilities, from (black-box use of) quantum-hard one-way functions [26]. The latter can then be used to obtain secure two-party *quantum* computation, by relying on the work of [17]. Quantum OT can also be used to obtain *multi-party* secure computation of all classical functionalities, in the presence of quantum computation and communication capabilities, and additionally assuming the existence of authenticated channels. This follows from the techniques in [26, 12, 14, 25] which obtain classical MPC based on black-box use of any OT protocol. By relying on [15], this also implies multi-party secure *quantum* computation.

In summary, our main result implies that: (1) 2PQC can be obtained from (black-box use of) quantum-hard OWFs and (2) assuming the existence of authenticated channels, MPQC can be obtained from (black-box use of) quantum-hard OWFs.

This gives a potential separation between the complexity of cryptographic primitives in the classical and quantum worlds. In the former, (two-party) secure computation provably cannot be based on black-box use of quantum-hard one-way functions. It is only known from special types of enhanced public-key encryption schemes or from the hardness of specific problems, both of which are believed to be much stronger assumptions than one-way functions. But in the quantum world, prior to our work, (two-party) secure computation was only known from the special commitments required in the protocol of [13], which can be based on QLWE following [13], or post-quantum OT (implicit in [20, 7, 2]) — but were not known to be achievable from quantum-hard one-way functions.

On the Significance of the Black-Box use of Cryptography in the Quantum Setting. Making black-box use of a cryptographic primitive refers to only having oracle access to its input/output behavior, without having the ability to examine the actual code (i.e., representation as a sequence of symbols) of the primitive. For instance, proving in zero-knowledge that a committed value satisfies some given predicate often requires explicit knowledge of the commitment algorithm; classifying the resulting proof as making "non-black-box" use of the one-way function. In the literature, constructions that make black-box use of cryptographic primitives are often preferred over those that make non-black-box use of cryptography. Besides their conceptual simplicity and elegance, black-box constructions are also of practical interest since they avoid expensive NP reductions involving circuits of primitives. Perhaps most importantly, in the case of black-box constructions, one can instantiate the underlying primitive with an arbitrary implementation, including physical implementations via secure hardware or those involving quantum communication.

In many quantum protocols, which involve quantum states being transferred between two or more players, black-box constructions are not only significantly preferable but often become *a necessity*. Let us illustrate this with an example. The GMW protocol [18] first showed that secure multi-party computation can be based on any oblivious transfer protocol; however the protocol involved zero-knowledge proofs involving the description of the (classical) oblivious transfer protocol. Due to the GMW [18] protocol being non-black-box in the underlying OT, our OT protocols cannot be used with GMW to obtain multi-party computation of classical functionalities. We instead need to rely on compilers like [26, 12, 14, 25] that only make *black-box use* of the underlying OT protocol. As discussed above, the black-box nature of these compilers makes them applicable irrespective of whether they are instantiated with classical or QOT.

In a similar vein, we believe that our *black-box* use of any statistically binding, quantum computationally hiding commitment in our QOT protocol is of particular significance. For instance, one can substitute our statistically binding, quantum computationally hiding commitment with an unconditionally secure one in the quantum random oracle model [34], resulting in *unconditional quantum OT in the quantum random oracle model.* Moreover if in the future, new constructions of statistically binding, quantum computationally hiding commitments involving quantum communication are discovered based on assumptions weaker than quantum-hard one-way functions, it would be possible to plug those into our protocol compilers to obtain QOT. These applications would not have been possible had we required non-black-box use of the underlying commitment.

Primary Tool: Stand-alone Extractable and Equivocal Commitments. As discussed earlier, [13] show that simulation-secure QOT can be obtained from commitments satisfying certain properties, namely *extraction* and *equivocation.*

- At a high level, extraction requires that there exist an efficient quantum "extractor" that is able to extract a committed message from any quantum committer.
- Equivocality requires that there exist an efficient quantum "equivocator" capable of simulating an interaction with any quantum receiver such that it can later open the commitment to any message of its choice.

These two properties are crucial for proving simulation security of the [11] OT protocol: extraction implies receiver security and equivocality implies sender security[1]. Our key technical contribution is the following:

Extractable and equivocal commitments can be based on the black-box use of quantum-hard one-way functions.

We obtain this result via the following transformations, each of which only makes *black-box* use of the underlying primitives.

– *Step 1: Quantum Equivocal Commitments from Quantum-Hard One-Way Functions.* We describe a generic unconditional compiler to turn any commitment into an equivocal commitment in the plain model. By applying our compiler to Naor's statistically binding commitment [32] — which can be based on quantum-hard one-way functions — we obtain a statistically binding, equivocal commitment.
– *Step 2: Quantum Extractable Commitments from Quantum Equivocal Commitments.* We show that the [11, 13, 8] framework can be used to obtain an extractable commitment that leverages quantum communication, and can be based on the existence of any quantum equivocal commitment. This combined with the previous step implies the existence of quantum extractable commitments based on the existence of quantum-hard one-way functions.
  This is in contrast to existing approaches (eg., [20]) that require classical communication but rely on qualitatively stronger assumptions like classical OT with post-quantum security.
– *Step 3: From Extractable Commitments to Extractable and Equivocal Commitments.* We apply the black-box equivocality compiler from the first step to the quantum extractable commitment obtained above, to produce an extractable and equivocal commitment.
  We point out that it is generally straightforward to make a classical commitment equivocal using zero-knowledge proofs, but this approach does not apply to quantum commitment protocols. We therefore devise our own equivocality compiler capable of handling quantum commitments and use it in both Step 1 and Step 3.

Plugging our quantum extractable and equivocal commitments into the [11] framework yields a final QOT protocol with an interaction pattern that readers familiar with [5, 11] may find interesting: the sender sends the receiver several BB84 states, after which the receiver proves to the sender that it has honestly measured the sender's BB84 states by *generating more BB84 states of their own and asking the sender to prove that they have measured the receiver's BB84 states.* An intriguing open question is whether obtaining QOT from one-way

[1] It is important to note that extraction and equivocation are only made possible in an ideal world where a simulator has access to the adversary's state. Participants in the real protocol cannot access each others' state, which prevents them from extracting or equivocating.

functions *requires* this type of two-way quantum communication or, alternatively, quantum memory.[2]

1.2 Related Work

For some readers, it may appear that the central claim of this work — that quantum-hard one-way functions suffice for oblivious transfer — has already been established [13, 8]. Indeed, prior work [13, 8] showed that statistically binding and computational hiding commitments (which are weaker than extractable and equivocal commitments), known to exist from one-way functions, can be plugged into the [11] template to achieve an oblivious transfer protocol satisfying *indistinguishability-based* security.

However, the indistinguishability-based security definition for oblivious transfer is *not* standard in the cryptographic literature. When cryptographers refer to "oblivious transfer", they almost always mean the standard *simulation-based* security notion. Indeed, the fundamental importance of oblivious transfer in modern cryptography is due to the fact that it is necessary and sufficient for secure computation, but this is only true for the simulation-based notion.

1.3 Concurrent and Independent Work

In a concurrent and independent work, Grilo, Lin, Song, and Vaikuntanathan [19] also construct simulation-secure quantum oblivious transfer from quantum-hard one way functions via the intermediate primitive of extractable and equivocal commitments. However, the two works take entirely different approaches to constructing these commitments. We briefly summarize these strategies below.

<u>This work:</u>

1. Construct equivocal commitments from statistically binding commitments via a new "equivocality compiler" based on Watrous [35] rewinding.
2. Construct extractable commitments from equivocal commitments via a new "extractability compiler" based on the [11] template.
3. Construct extractable and equivocal commitments from extractable commitments via the same compiler from Step 1.

<u>[19]</u>:

1. Construct selective opening secure commitments with inefficient simulation against malicious committers from statistically binding commitments and zero-knowledge proofs.
2. Construct QOT with inefficient simulation against malicious receivers from selective opening secure commitments with inefficient simulation against malicious committers, following the [11] QOT template.[3]

---

[2] Naive approaches to removing one direction of quantum communication appear to require the honest parties to be entangled and subsequently perform quantum teleportation.

[3] [19] point out that the conclusions of Steps 1 and 2 together had also been established in prior works of [33, 13, 8].

3. Construct parallel QOT with inefficient simulation against malicious receivers from (stand-alone) QOT with inefficient simulation against malicious receivers via a new lemma for parallel repetition of protocols.
4. Construct verifiable conditional disclosure of secrets, a new primitive introduced in [19], from parallel QOT with inefficient simulation against malicious receivers, statistically binding commitments, Yao's garbled circuits, and zero-knowledge proofs.
5. Construct extractable commitments from verifiable conditional disclosure of secrets, statistically binding commitments, and zero-knowledge proofs.
6. Construct extractable and equivocal commitments from extractable commitments and zero-knowledge proofs.

We believe that our result is easier to understand and conceptually simpler, as we do not need to define additional primitives beyond extractable and/or equivocal commitments. Aside from differences in approach, there are several other places where the results differ:

– **This Work: Black-Box Use of One-Way Functions.** A significant advantage of our work over [19] is that we construct quantum OT from *black-box use* of statistically binding commitments or one-way functions. The OT in [19] makes non-black-box use of the underlying one-way function. As discussed above, making black-box use of underlying cryptographic primitives is particularly useful in the quantum setting. Due to the extensive use of zero-knowledge proofs and garbled circuits in [19], it appears difficult to modify their approach to be black-box in the underlying one-way function.
– **This Work: One-Sided Statistical Security.** Additionally, our oblivious transfer protocol offers *one-sided statistical security*. As written, our quantum OT protocol satisfies statistical security against malicious senders (and computational security against malicious receivers). Moreover, this OT can be reversed following the techniques in eg., [36] to obtain a quantum OT protocol that satisfies statistical security against malicious receivers (and computational security against malicious senders). On the other hand, the quantum OT protocol in [19] appears to achieve computational security against both malicious senders and malicious receivers.
– **[19]: Verifiable Conditional Disclosure of Secrets.** Towards achieving their main result, [19] introduce and construct verifiable conditional disclosure of secrets (vCDS). This primitive may be of independent interest.
– **[19]: Constant Rounds in the CRS Model.** While both works construct poly(λ)-round protocols in the plain model, [19] additionally construct a constant round OT protocol in the CRS model based on (non-black-box use of) quantum-hard one-way functions.
In an earlier version of this work, we did not consider the CRS model. After both works were posted to the Cryptology ePrint Archive, we realized that our techniques could be straightforwardly adapted to achieve constant round complexity in the CRS model, while still making black-box use of one-way functions. However, unlike [19], our CRS is non-reusable. For the interested reader, we sketch how this can be achieved in the full version.

2 Technical Overview

This work establishes that (1) black-box use of post-quantum one-way functions suffices for post-quantum *extractable and equivocal* commitment schemes and moreover, that (2) [11] quantum oblivious transfer instantiated with such commitments is a standard *simulation-secure* oblivious transfer. Crucially, the standard notion of simulation-secure (quantum) oblivious transfer that we achieve is sequentially composable and suffices to achieve general-purpose secure quantum computation. Before explaining our technical approach, we provide a complete review of the original [11] protocol.

2.1 Recap: Quantum Oblivious Transfer from Commitments

In quantum oblivious transfer (QOT), a quantum sender holding two classical messages m_0, m_1 engages in an interactive protocol over a quantum channel with a quantum receiver holding a classical choice bit b. Correctness requires the receiver to learn m_b by the end of the protocol. Informally, security demands that a malicious receiver only learn information about one of m_0, m_1, and that a malicious sender learn nothing about b. Somewhat more formally, as discussed earlier, our focus is on the standard simulation-based notion of security. This stipulates the existence of an efficient quantum simulator that generates the view of an adversary (sender/receiver) when given access to an ideal OT functionality. In particular, when simulating the view of a malicious sender, this simulator must extract the sender's inputs (m_0, m_1) without knowledge of the receiver's input b. And when simulating the view of a malicious receiver, the simulator must extract the receiver's input b, and then simulate the receiver's view given just m_b.

We recall the construction of quantum oblivious transfer due to [11] (henceforth CK88), which combines the information theoretic quantum key distribution protocol of [5] (henceforth BB84) with cryptographic bit commitments.

CK88 First Message. The first message of the CK88 protocol exactly follows the beginning of the BB84 protocol. For classical bits y, z, let $|y\rangle_z$ denote $|y\rangle$ if $z = 0$, and $(|0\rangle + (-1)^y |1\rangle)/\sqrt{2}$ if $z = 1$, i.e. the choice of z specifies whether to interpret y as a computational or Hadamard basis vector. Let λ denote the security parameter. The sender samples two random 2λ-bit strings x and θ, and constructs "BB84 states" $|x_i\rangle_{\theta_i}$ for $i \in [2\lambda]$. The sender forwards these 2λ BB84 states $(|x_i\rangle_{\theta_i})_{i\in[2\lambda]}$ to the receiver. Next, the receiver samples a 2λ-bit string $\hat{\theta}$, measures each $|x_i\rangle_{\theta_i}$ in the basis specified by $\hat{\theta}_i$, and obtains a 2λ-bit measurement result string $\hat{x}$.

CK88 Measurement-Check Subprotocol. At this point, the CK88 and BB84 protocols diverge. Since the BB84 protocol is an interaction between two *honest* parties, it assumes the parties comply with the protocol instructions. However, in the CK88 protocol, a malicious receiver who does not measure these BB84 states will be able to compromise sender privacy later in the protocol. Therefore, the next phase of CK88 is a measurement-check subprotocol designed to catch a malicious receiver who skips the specified measurements. This subprotocol requires the use of a quantum-secure classical commitment scheme; for the purposes of this recap, one should imagine a commitment with idealized hiding and binding properties. The subprotocol proceeds as follows:

- For each $i \in [2\lambda]$, the receiver commits to $(\hat{\theta}_i, \hat{x}_i)$.
- Next, the sender picks a random set T of λ indices from $[2\lambda]$, and challenges the receiver to open the corresponding commitments.
- The receiver sends $(\hat{\theta}_i, \hat{x}_i)$ along with the corresponding opening for each $i \in T$.
- The sender verifies each commitment opening, and furthermore checks that $\hat{x}_i = x_i$ for each $i \in T$ where $\hat{\theta}_i = \theta_i$. If any of these checks fail, the sender aborts.

The rough intuition for the subprotocol is simple: from the receiver's point of view, the BB84 states are maximally mixed and therefore completely hide x_i and θ_i. For any index i that the receiver does not measure, it must guess $\hat{x}_i$. From the receiver's perspective, the sender checks $\hat{x}_i$ against x_i if two 1/2-probability events occur: (1) i is included in T, and (2) $\hat{\theta}_i = \theta_i$. This means a malicious receiver who skips a significant number of measurements will be caught with overwhelming probability.

CK88 Privacy Amplification. If all the subprotocol checks pass, the sender continues to the final stage of the CK88 protocol. For convenience, relabel the λ indices in $[2\lambda] \setminus T$ from 1 to λ; all indices corresponding to opened commitments are discarded for the remainder of the protocol.

For each $i \in [\lambda]$, the sender reveals the correct measurement basis θ_i. The receiver then constructs the index set I_b — where b is its choice bit for the oblivious transfer — as the set of all $i \in [\lambda]$ where $\theta_i = \hat{\theta}_i$. It sets I_{1-b} to be the remaining indices, and sends (I_0, I_1) to the sender. Note that by the hiding property of the commitments, the sender should not be able to deduce b from (I_0, I_1); furthermore, I_0 and I_1 will both be close to size $\lambda/2$, since for each $i \in [\lambda]$, the receiver committed to $\hat{\theta}_i$ before obtaining θ_i.

On receiving I_0, I_1, the sender sets $x_0 \coloneqq (x_i)_{i \in I_0}$ and $x_1 \coloneqq (x_i)_{i \in I_1}$. The intuition is that if a receiver honestly constructs (I_0, I_1), it will only have information about x_b corresponding to its choice bit b. However, it turns out that even if the receiver maliciously constructs (I_0, I_1), at least one of x_0 and x_1 will have high min-entropy from its point of view. Thus, by standard privacy amplification techniques, the sender can complete the oblivious transfer as follows. It samples two universal hash functions h_0 and h_1, both with ℓ-bit outputs, and uses $h_0(x_0)$ to mask the ℓ-bit message m_0, and uses $h_1(x_1)$ to mask m_1. That is, the sender sends $(h_0, h_1, h_0(x_0) \oplus m_0, h_1(x_1) \oplus m_1)$ to the receiver, who can then use x_b to recover m_b. Since x_{1-b} will have high entropy, the leftover hash lemma implies that $h_{1-b}(x_{1-b})$ is statistically close to uniform, which hides m_{1-b} from the receiver.

Simulation-Based Security. Turning this intuition into a proof of simulation-based security of the resulting QOT requires some additional insights [13], and requires the commitments used in the measurement-check subprotocol to satisfy two additional properties: extractability and equivocality. In what follows, we briefly summarize why these properties help achieve simulation-based security.

To argue that the resulting QOT protocol satisfies security against a malicious sender, one must demonstrate the existence of a simulator that simulates the sender's view by generating messages on behalf of an honest receiver, and extracts both QOT inputs of the sender[4]. Now, the measurement-check subprotocol described above is designed to ensure that at least one of the sender's inputs is hidden from a receiver. To nevertheless enable the simulator to extract both sender inputs, the idea in [13] is to modify the commitments used in the measurement-check subprotocol with *equivocal commitments* that allow the simulator to later open these commitments to any value of its choice. This enables the simulator to defer any measurements until after it obtains the set T from the sender, and then selectively measure *only* the states that correspond to indices in T. All other states are left untouched until the sender reveals its measurement bases in the final stage of the CK88 protocol. Upon obtaining the sender's "correct" measurement bases, the simulator measures all the remaining states in the correct bases, allowing it to learn both the inputs of the sender.

To demonstrate that the resulting QOT protocol satisfies security against a malicious receiver, one must demonstrate the existence of a simulator that simulates the receiver's view by generating messages on behalf of an honest sender, and extracts the receiver's choice bit. Again by design, the measurement-check subprotocol ensures that the receiver's choice bit hidden is hidden from the sender. To nevertheless enable the simulator to extract this choice bit, [13] modify the commitments in the measurement-check subprotocol so that the simulator is able to extract all of the $\{(\widehat{\theta}_i, \widehat{x}_i)\}_{i\in[2\lambda]}$ from the receiver's commitments. This enables the simulator to compute which one of the sets I_0, I_1 contain more indices i for which $\theta_i = \widehat{\theta}_i$; clearly the set with more indices corresponds to the receiver's choice bit. In summary, the key tool that enables simulation against a malicious receiver is an *extractable* commitment, that forces the receiver to use commitments for which the simulator can extract the committed value, without ever running the opening phase.

To conclude, following [13] the CK88 protocol can be shown to satisfy simulation-based security as long as the commitments used in the measurement-check subprotocol satisfy both the *extractability* and *equivocality* properties that were informally described above.

With this in mind, we now describe our primary technical contribution: a construction of the required extractable and equivocal commitments based on black-box use of quantum-hard one-way functions.

[4] We refer the reader to Section 6.1 for a formal definition of simulation-based QOT.

2.2 Our Construction: A High-Level Overview

The rest of this technical overview describes our *black-box* construction of simultaneously *extractable and equivocal* quantum bit commitments from any quantum-hard one-way function.

The ingredients for our construction are the following:

- A general-purpose "equivocality compiler" that turns any bit commitment scheme — classical or quantum — into an *equivocal* quantum commitment scheme. Moreover, if the original commitment scheme is *extractable*, this compiler outputs an *extractable and equivocal* commitment scheme.
- A general-purpose "extractability compiler" that turns any *equivocal* bit commitment scheme — classical or quantum — into an *extractable but not equivocal* commitment scheme.

Both of these compilers require no additional computational assumptions beyond those of the original commitment schemes. Given these compilers, we build extractable and equivocal commitments via the following steps:

- **Instantiation:** Begin with Naor's statistically-binding, computationally hiding commitments [32]. Naor's construction makes black-box use of one-way functions and achieves post-quantum computational hiding assuming post-quantum security of the one-way function.[5]
- **Step 1:** Plug Naor's commitments into our equivocality compiler to obtain an *equivocal* quantum bit commitment scheme.
- **Step 2:** Feed the resulting equivocal quantum bit commitments into our extractability compiler to obtain an *extractable but not equivocal* quantum bit commitment.
- **Step 3:** Run the equivocality compiler *a second time*, but now starting with the extractable commitments produced by the previous step. This gives the desired *extractable and equivocal* quantum bit commitments.

2.3 Making Any Quantum (or Classical) Commitment Equivocal

Recall that a quantum commitment protocol is *equivocal* if an efficient quantum algorithm called the *equivocator*, with access to the receiver, can generate commitments that can be opened to any value. More precisely, for any receiver (mod-

[5] In slightly more detail, Naor's commitment scheme makes black-box use of any pseudo-random generator (PRG). It is straightforward to verify that if the PRG is post-quantum secure, the commitment satisfies computational hiding against quantum attackers. A black-box construction of pseudo-random generators from one-way functions is due to [21]; Aaronson [1] and Zhandry [38] observed that [21] applies to non-uniform quantum attackers with *classical* advice. This can be extended to handle non-uniform quantum advice by giving the one-way function attacker constructed in the [21] reduction many copies of the PRG attacker's non-uniform quantum advice (which only requires some polynomial upper bound on the number of times the reduction invokes the PRG attacker).

eled as an efficient malicious quantum algorithm), there must exist an equivocator who can generate a computationally indistinguishable commitment that the equivocator can later open arbitrarily.

In this subsection, we describe a *black-box* compiler for a fairly general task (which may be of independent interest): making any *classical or quantum* commitment equivocal. Recall from Section 2.2 that we will need to invoke our equivocality compiler *twice*, once on a classical bit commitment scheme, and once on an extractable quantum bit commitment scheme; in the latter case, our compiler will need to preserve the extractability of the original commitment. Since classical commitments are a subclass of quantum commitments, our exposition will focus on challenges unique to the quantum setting.

Our Equivocality Compiler. In our construction, to commit to a bit b, the committer and receiver will perform λ sequential repetitions of the following subprotocol:

– The (honest) committer samples 2 uniformly random bits u_0, u_1, and commits to each one *twice* using the base commitment scheme. Let the resulting commitments be $\mathbf{c}_0^{(0)}, \mathbf{c}_0^{(1)}, \mathbf{c}_1^{(0)}, \mathbf{c}_1^{(1)}$, where the first two are to u_0 and the second two are to u_1. Note that since the base commitment scheme can be an arbitrary quantum interactive commitment, each commitment $\mathbf{c}_{b_1}^{(b_2)}$ corresponds to the receiver's quantum state after the commitment phase of the base commitment.
– The receiver sends the committer a random challenge bit β.
– The committer opens the two base commitments $\mathbf{c}_\beta^{(0)}, \mathbf{c}_\beta^{(1)}$. If the openings are invalid or the revealed messages are different, the receiver aborts the entire protocol.

If these λ executions pass, the receiver is convinced that a majority of the committer's remaining 2λ unopened commitments are honestly generated, i.e. most pairs of commitments are to the same bit.

Rewriting the (honest) committer's unopened bits as $u_1, \ldots, u_\lambda$, the final step of the commitment phase is for the committer to send $h_i \coloneqq u_i \oplus b$ for each $i \in [\lambda]$ (recall that b is the committed bit).

To decommit, the committer reveals each u_i by picking one of the two corresponding base commitments at random, and opening it. The receiver accepts if each one of the base commitment openings is valid, and the opened u_i satisfies $h_i \oplus u_i = b$ for every i.

The (statistical) binding property of the resulting commitment can be seen to follow from the (statistical) binding of the underlying commitment. For any commitment, define the unique committed value as the majority of $(h_i \oplus u_i)$ values in the unopened commitments, setting u_i to $\perp$ if both committed bits in the i^{th} session differ. Due to the randomized checks by the receiver, any committer that tries to open to a value that differs from the unique committed value will already have been caught in the commit phase, and the commitment will have been rejected with overwhelming probability. A similar argument also allows us to establish that this transformation preserves extractability of the underlying commitment. We now discuss why the resulting commitment is *equivocal*.

Quantum Equivocation. The natural equivocation strategy should have the equivocator (somehow) end up with λ pairs of base commitments where for each $i \in [\lambda]$, the pair of commitments is to u_i and $1 - u_i$ for some random bit u_i. This way, it can send an appropriately distributed string $h_1, \cdots, h_\lambda$, and later open to any b by opening the commitment to $b \oplus h_i$ for each i.

We construct our equivocator using Watrous's quantum rewinding lemma [35] (readers familiar with Watrous's technique may have already noticed our construction is tailored to enable its use).

We give a brief, intuition-level recap of the rewinding technique as it pertains to our equivocator. Without loss of generality, the malicious quantum receiver derives its challenge bit β by performing some binary outcome measurement on the four quantum commitments it has just received (and on any auxiliary states). Our equivocator succeeds (in one iteration) if it can prepare four quantum commitments $\mathbf{c}_0^{(0)}, \mathbf{c}_0^{(1)}, \mathbf{c}_1^{(0)}, \mathbf{c}_1^{(1)}$ where:

1. $\mathbf{c}_\alpha^{(0)}, \mathbf{c}_\alpha^{(0)}$ are commitments to the same random bit,
2. $\mathbf{c}_{1-\alpha}^{(0)}, \mathbf{c}_{1-\alpha}^{(0)}$ are commitments to a random bit and its complement,
3. on input $\mathbf{c}_0^{(0)}, \mathbf{c}_0^{(1)}, \mathbf{c}_1^{(0)}, \mathbf{c}_1^{(1)}$, the receiver produces challenge bit $\beta = \alpha$.

That is, the equivocator is successful if the receiver's challenge bit β corresponds to the bit α that it can open honestly. Watrous's [35] rewinding lemma applies if the distribution of β is *independent* of the receiver's choice of α, which is guaranteed here by the hiding of the base commitments. Thus, the rewinding lemma yields a procedure for obtaining an honest-looking interaction where all three properties above are met. Given the output of the rewinding process, the equivocator has successfully "fooled" the committer on this interaction and proceeds to perform this for all λ iterations. As described above, fooling the committer on all λ iterations enables the equivocator to later open the commitment arbitrarily.

2.4 An Extractability Compiler for Equivocal Commitments

In this subsection, we compile any classical or quantum *equivocal* bit commitment into a quantum *extractable* bit commitment. We stress that even though this compiler is applied to equivocal bit commitments, the resulting commitment is *not* guaranteed to be simultaneously *extractable and equivocal*; we refer the reader to Section 2.2 for details on how this compiler fits into our final construction. Recall that a commitment scheme is *extractable* if for any adversarial quantum committer that successfully completes the commitment phase, there exists an efficient quantum algorithm (called the extractor) which outputs the committed bit.

Construction. The committer, who intends to commit to a classical bit b, begins by sampling 2λ-bit strings x and θ. It generates the corresponding 2λ BB84 states $|x_i\rangle_{\theta_i}$ and sends this to the receiver. The receiver picks 2λ random measurement bases $\hat{\theta}_i$, and measures each $|x_i\rangle_{\theta_i}$ in the corresponding basis, obtaining outcomes $\hat{x}_i$.

Next, the receiver and committer engage in a CK88-style measurement-check subprotocol. That is, they temporarily switch roles (for the duration of the subprotocol), and perform the following steps:

1. The receiver (acting as a committer in the subprotocol), commits to each $\hat{\theta}_i$ and $\hat{x}_i$ (for each $i \in [2\lambda]$) with an *equivocal* commitment.
2. The committer (acting as a receiver in the subprotocol), asks the receiver to open the equivocal commitments for all $i \in T$, where $T \subset [2\lambda]$ is a random set of size λ.
3. The receiver (acting as a committer in the subprotocol) opens the λ commitments specified by T.

Provided the receiver passes the measurement-check subprotocol, the committer generates the final message of the commitment phase as follows:

- Discard the indices in T and relabel the remaining λ indices from 1 to λ.
- Partition $\{x_1, \ldots, x_\lambda\}$ into $\sqrt{\lambda}$ strings $\vec{x}_1, \ldots, \vec{x}_{\sqrt{\lambda}}$ each of length $\sqrt{\lambda}$.
- Sample $\sqrt{\lambda}$ universal hash functions $h_1, \ldots, h_{\sqrt{\lambda}}$ each with 1-bit output.
- Finally, send $(\theta_i)_{i\in[\lambda]}, (h_j, h_j(\vec{x}_j) \oplus b)_{j\in[\sqrt{\lambda}]}$.

This concludes the commitment phase.

To decommit, the committer reveals b and $(\vec{x}_1, \ldots, \vec{x}_{\sqrt{\lambda}})$. The receiver accepts if (1) for each j, the bit b and the value $\vec{x}_j$ are consistent with the claimed value of $h_j(\vec{x}_j) \oplus b$ from the commit phase, and (2) for each index $i \in [\lambda]$ where $\theta_i = \hat{\theta}_i$, the x_i from the opening is consistent with $\hat{x}_i$.

Extraction. The use of equivocal commitments in the measurement-check subprotocol makes extraction simple. Given any malicious committer, we construct an extractor as follows.

The extractor plays the role of the receiver and begins an interaction with the malicious committer. But once the committer sends its 2λ BB84 states, the extractor skips the specified measurements, instead leaving these states unmeasured. Next, instead of performing honest commitments to each $\hat{\theta}_i, \hat{x}_i$, the extractor invokes (for each commitment) the equivocator algorithm of the underlying equivocal commitment scheme. Since the equivocator is guaranteed to produce an indistinguishable commitment from the point of view of any malicious receiver for the equivocal commitment, this dishonest behavior by the extractor will go undetected.

When the malicious committer responds with a challenge set $T \subset [2\lambda]$, the extractor samples uniformly random bases $\hat{\theta}_i$ for each $i \in T$, measures the corresponding BB84 states to obtain $\hat{x}_i$ values, and sends $(\hat{\theta}_i, \hat{x}_i)_{i\in T}$. Moreover, the equivocator (for each commitment) will enable the extractor to generate valid-looking openings for all of these claimed values.

Thus, the malicious committer proceeds with the commitment protocol, and sends

$$(\theta_i)_{i\in[\lambda]}, (h_j, h_j(\vec{x}_j) \oplus b)_{j\in[\sqrt{\lambda}]}$$

to the extractor. These correspond to the λ BB84 states that the extractor has not yet measured, so it can simply read off the bases θ_i, perform the specified measurements, and extract the committer's choice of b.

Statistical Hiding. Intuitively, statistical hiding of the above commitment protocol follows because the measurement-check subprotocol forces the receiver to measure states in arbitrary bases, which destroys information about the corresponding x_i values whenever $\widehat{\theta}_i \neq \theta_i$. The formal argument is a straightforward application of a quantum sampling lemma of [8], devised in part to simplify analysis of [11]-style protocols, and we defer further details to the supplementary materials.

2.5 Putting it Together: From Commitments to Secure Computation.

Plugging the compilers of Sections 2.3 and 2.4 into the steps described in Section 2.2 yields a black-box construction of simultaneously extractable and equivocal quantum bit commitments from quantum-hard one-way functions. Following [13], these commitments can be plugged into CK88 to obtain maliciously simulation-secure QOT (see Section 6 for further details). Finally, going from QOT to arbitrary secure computation (in a black-box way) follows from prior works of [26, 25, 17, 15]; a more thorough discussion is available in the supplementary materials.

3 Preliminaries

Notation. We will write density matrices/quantum random variables (henceforth, QRVs) in lowercase bold font, e.g. $\mathbf{x}$. A quantum register X will be written in uppercase (grey) serif font. A collection of (possibly entangled) QRVs will be written as $(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

Throughout this paper, λ will denote a cryptographic security parameter. We say that a function $\mu(\lambda)$ is *negligible* if $\mu(\lambda) = 1/\lambda^{\omega(1)}$.

The trace distance between two QRVs $\mathbf{x}$ and $\mathbf{y}$ will be written as $\|\mathbf{x} - \mathbf{y}\|_1$. Recall that the trace distance captures the maximum probability that two QRVs can be distinguished by any (potentially inefficient) procedure. We therefore say that two infinite collections of QRVs $\{\mathbf{x}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathbf{y}_\lambda\}_{\lambda\in\mathbb{N}}$ are *statistically indistinguishable* if there exists a negligible function $\mu(\lambda)$ such that $||\mathbf{x}_\lambda - \mathbf{y}_\lambda||_1 \leq \mu(\lambda)$, and we will frequently denote this with the shorthand $\{\mathbf{x}_\lambda\}_{\lambda\in\mathbb{N}} \approx_s \{\mathbf{y}_\lambda\}_{\lambda\in\mathbb{N}}$.

Non-Uniform Quantum Advice. We will consider non-uniform quantum polynomial-time (QPT) algorithms *with quantum advice*, denoted by $\mathcal{A} = \{\mathcal{A}_\lambda, \boldsymbol{\rho}_\lambda\}_{\lambda\in\mathbb{N}}$, where each $\mathcal{A}_\lambda$ is the classical description of a poly(λ)-size quantum circuit, and each $\boldsymbol{\rho}_\lambda$ is some (not necessarily efficiently computable) non-uniform poly(λ)-qubit quantum advice. We remark that "non-uniform quantum polynomial-time algorithms" often means non-uniform *classical advice*, but the cryptographic applications in this work will require us to explicitly consider quantum advice.

Definitions for Cryptographic Commitments. Full definitions of cryptographic commitments can be found in the supplementary materials.

4 A Quantum Equivocality Compiler

In this section, we show a generic black-box compiler that takes any quantum-secure bit commitment scheme and produces a quantum-secure *equivocal* bit commitment scheme.

The compiler is described in Protocol 1, where $(\mathsf{Commit}, \mathsf{Decommit})$ denotes some statistically binding and computationally hiding bit commitment scheme. We describe how to equivocally commit to a single bit, and note that commitment to an arbitrary length string follows by sequential repetition.

Furthermore, we show that if the underlying commitment $(\mathsf{Commit}, \mathsf{Decommit})$ is *extractable*, then the resulting commitment is both extractable and equivocal.

These results are captured in the following theorems.

Theorem 1. *For $\mathcal{X} \in \{$quantum extractability, statistical binding$\}$ and $\mathcal{Y} \in \{$computationally, statistically$\}$, if* Commit *is a $\mathcal{Y}$-hiding quantum bit commitment satisfying $\mathcal{X}$, then Protocol 1 is a $\mathcal{Y}$-equivocal bit commitment satisfying $\mathcal{X}$.*

These theorems follow from establishing statistical binding, equivocality, and extractability of the commitment in Protocol 1, as we do next. First, we note that if Commit is statistically binding, then Protocol 1 is statistically binding. For any adversarial committer strategy, consider the λ unopened pairs of commitments after the commit phase. Since Commit is statistically binding, we can assume that each of the 2λ commitments is binding to a particular bit, except with negligible probability. Now, if any single pair contains binding commitments to the same bit d_i, then the committer will only be able to open its Protocol 1 commitment to the bit $d_i \oplus e_i$. Thus, to violate binding, the adversarial committer will have to have committed to different bits in each of the λ unopened pairs. However, in this case, the committer will be caught and the receiver will abort except with probability $1/2^\lambda$.

4.1 Equivocality

The equivocal simulator $(\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}, \mathcal{Q}_{\mathcal{R}^*,\mathsf{open}})$ is obtained via the use of Watrous's quantum rewinding lemma [35]; a full statement of the lemma is available in the

Protocol 1

Committer $\mathcal{C}$ Input: Bit $b \in \{0,1\}$.
<u>The Protocol: Commit Phase</u>

1. $\mathcal{C}$ samples uniformly random bits $d_{i,j}$ for $i \in [\lambda]$ and $j \in \{0,1\}$.
2. For every $i \in [\lambda]$, $\mathcal{C}$ and $\mathcal{R}$ sequentially perform the following steps.
   (a) $\mathcal{C}$ and $\mathcal{R}$ execute four sessions sequentially, namely:
      – $\mathbf{x}_{0,0}, \mathbf{y}_{0,0} \leftarrow \mathsf{Commit}\langle \mathcal{C}(d_{i,0}), \mathcal{R}\rangle$,
      – $\mathbf{x}_{0,1}, \mathbf{y}_{0,1} \leftarrow \mathsf{Commit}\langle \mathcal{C}(d_{i,0}), \mathcal{R}\rangle$,
      – $\mathbf{x}_{1,0}, \mathbf{y}_{1,0} \leftarrow \mathsf{Commit}\langle \mathcal{C}(d_{i,1}), \mathcal{R}\rangle$ and
      – $\mathbf{x}_{1,1}, \mathbf{y}_{1,1} \leftarrow \mathsf{Commit}\langle \mathcal{C}(d_{i,1}), \mathcal{R}\rangle$.
   (b) $\mathcal{R}$ sends a choice bit $c_i \leftarrow \{0,1\}$.
   (c) $\mathcal{C}$ and $\mathcal{R}$ execute two decommitments, obtaining the opened bits:
      – $u \leftarrow \mathsf{Decommit}\langle \mathcal{C}(\mathbf{x}_{c_i,0}), \mathcal{R}(\mathbf{y}_{c_i,0})\rangle$ and
      – $v \leftarrow \mathsf{Decommit}\langle \mathcal{C}(\mathbf{x}_{c_i,1}), \mathcal{R}(\mathbf{y}_{c_i,1})\rangle$.

      If $u \neq v$, $\mathcal{R}$ aborts. Otherwise, $\mathcal{C}$ and $\mathcal{R}$ continue.
3. For $i \in [\lambda]$, $\mathcal{C}$ sets $e_i = b \oplus d_{i,1-c_i}$ and sends $\{e_i\}_{i\in[\lambda]}$ to R.

<u>The Protocol: Decommit Phase</u>

1. $\mathcal{C}$ sends b to $\mathcal{R}$. In addition,
   – For $i \in [\lambda]$, $\mathcal{C}$ picks $\alpha_i \leftarrow \{0,1\}$ and sends it to $\mathcal{R}$.
   – $\mathcal{C}$ and $\mathcal{R}$ execute $\widehat{d_i} \leftarrow \mathsf{Decommit}\langle \mathcal{C}(\mathbf{x}_{1-c_i,\alpha_i}), \mathcal{R}(\mathbf{y}_{1-c_i,\alpha_i})\rangle$.
2. $\mathcal{R}$ accepts the decommitment and outputs b if for every $i \in [\lambda]$, $\widehat{d_i} = b \oplus e_i$.

Fig. 1. Equivocal Bit Commitment.

supplementary materials. For the purposes of defining the simulation strategy, it will be sufficient (w.l.o.g.) to consider a restricted receiver $\mathcal{R}^*$ as follows, for the i^{th} sequential step of the protocol. In our simulation, the state of $\mathcal{R}^*$ will be initialized to the final state at the end of simulating the $(i-1)^{th}$ step.

1. $\mathcal{R}^*$ takes a quantum register W, representing its auxiliary quantum input. $\mathcal{R}^*$ will use two additional quantum registers that function as work space: V, which is an arbitrary (polynomial-size) register, and A, which is a single qubit register. The registers V and A are initialized to their all-zero states before the protocol begins.
2. Let M denote the polynomial-size register used by $\mathcal{C}$ to send messages to $\mathcal{R}^*$. After carrying out step 2(a) by running on registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$, $\mathcal{R}^*$ measures the register A to obtain a bit c_i, for Step 2(b), which it sends back to $\mathcal{C}$.
3. Next, $\mathcal{R}^*$ computes the decommitment phases (with messages from $\mathcal{C}$ placed in register M) according to Step 2(c). $\mathcal{R}^*$ outputs registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$.

Any polynomial-time quantum receiver can be modeled as a receiver of this restricted form followed by some polynomial-time post-processing of the restricted receiver's output. The same post-processing can be applied to the output of the simulator that will be constructed for the given restricted receiver.

Following [35], we define a simulator that uses two additional registers, C and Z. C is a one qubit register, while Z is an auxiliary register used to implement the computation that will be described next. Consider a quantum procedure $\mathcal{Q}_{\mathsf{partial}}$ that implements the strategy described in Protocol 2 using these registers.

Protocol 2

Circuit $\mathcal{Q}_{\mathsf{partial}}$**.**

1. Sample a uniformly random classical bit $\widehat{c}$, and store it in register C.
2. Sample uniformly random bits (z, d).
3. If $\widehat{c} = 0$, initialize committer input as follows, corresponding to four sequential sessions:
   - For the first two sessions, set committer input to z.
   - For the third and fourth sessions, set committer input to d and $1 - d$ respectively.
4. If $\widehat{c} = 1$, initialize committer input as follows, corresponding to four sequential sessions:
   - For the first and second sessions, set committer input to d and $1 - d$ respectively.
   - For the last two sessions, set committer input to z.
5. Run the commitment phase interaction between the honest committer and $\mathcal{R}^*$'s sequence of unitaries on registers initialized as above.
6. Measure the qubit register A to obtain a bit c. If $c = \widehat{c}$, output 0, otherwise output 1.

Fig. 2. Equivocal Simulator.

Next, we would like to apply Watrous's quantum rewinding lemma to the $\mathcal{Q}_{\mathsf{partial}}$ circuit. In order to do this, we will argue that the probability $p(\psi)$ that this circuit outputs 0 is such that $|p(\psi) - \frac{1}{2}| = \mathsf{negl}(\lambda)$, regardless of the auxiliary input $|\psi\rangle$ to $\mathcal{R}^*$. This follows from the fact that the commitments are (statistically/computationally) hiding. In more detail, by definition, Step 5 produces a distribution on the $\mathcal{R}^*$'s side that is identical to the distribution generated by $\mathcal{R}^*$ in its interaction with the committer. If $|p(\psi) - \frac{1}{2}|$ were non-negligible, then the sequence of unitaries applied by $\mathcal{R}^*$ could be used to distinguish commitments generated according to the case $\widehat{c} = 0$ from commitments generated according to the case $\widehat{c} = 1$, leading to a contradiction.

Now consider the state of the residual qubits of $\mathcal{Q}_{\mathsf{partial}}$ conditioned on a measurement of its output qubit being 0. The output state of the general quantum circuit $\widehat{\mathcal{Q}}$ resulting from applying Watrous's quantum rewinding lemma will have negligible trace distance from this state. This state is over all of the registers discussed above, so the simulator $\mathcal{Q}_{\mathsf{com},\mathcal{R}^*}$ must further process this state as:

- Measure the register C, obtaining challenge c.
- Compute decommitment information corresponding to challenge c, as in Step 2(c) of the protocol (recall that this information is stored in the message register M).
- Output registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$. All remaining registers are traced out.

The simulator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}$ executes all i sequential interactions in this manner, and then samples $e_1, \ldots, e_\lambda \leftarrow \{0,1\}^\lambda$, as the committer messages for Step 3 of Protocol 1. It runs the receiver's unitary on the resulting protocol, and outputs the resulting registers $(\mathsf{W}, \mathsf{V}, \mathsf{A}, \mathsf{M})$. It additionally outputs private state $\mathsf{st} = (c_1, d_1, \ldots, c_\lambda, d_\lambda)$ where c_i, d_i were sampled during the ith execution of Protocol 2.

The simulator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{open}}(b, \mathsf{st}, \mathbf{w}, \mathbf{v}, \mathbf{a}, \mathbf{m})$ parses st as $(c_1, d_1, \ldots, c_\lambda, d_\lambda)$. For every $i \in [\lambda]$ it does the following:

- Let $\widehat{d_i} = b \oplus e_i$.
- If $c_i = 0$, it executes the decommitment phase for the $((\widehat{d_i} \oplus d_i) + 2)^{th}$ session.
- If $c_i = 1$, it executes the decommitment phase for the $(\widehat{d_i} \oplus d_i)^{th}$ session.

$\mathcal{Q}_{\mathcal{R}^*,\mathsf{open}}$ then executes the receiver's algorithm on these decommitments and outputs the resulting state. Note that each decommitment will be to the bit $\widehat{d_i} = b \oplus e_i$.

To complete the proof of equivocality, we must establish that the view of the receiver interacting with an honest committer the view of the receiver interacting with the equivocator are indistinguishable. This follows from the (statistical/computational) hiding of the commitment scheme, via an identical argument to the one used above. In particular, if the equivocal simulator produces a distribution that is distinguishable from the real distribution, then there exists a session $i \in [\lambda]$ such that the distribution in the real and ideal experiments upto the $i-1^{th}$ session are indistinguishable, but upto the i^{th} session are distinguishable. This contradicts the above guarantee given by the quantum rewinding lemma, since for any i, the post-processed residual qubits of $\mathcal{Q}_{\mathsf{partial}}$ are indistinguishable from the state of $\mathcal{R}^*$ after the i^{th} sequential session in the real protocol (due to the hiding of the commitment scheme).

4.2 Extractability

Next, we prove that Protocol 1 satisfies extractability as long as the underlying commitment $(\mathsf{Commit}, \mathsf{Decommit})$ is extractable; in other words, this compiler preserves extractability. Consider the following extractor $\mathcal{E}_{\mathcal{C}^*}$.

- For $i \in [\lambda]$:

- Execute four sequential commitment sessions with $\mathcal{C}^*$, where the extractor of Commit is run on all sessionss. Obtain outputs $(\boldsymbol{\rho}_{\mathcal{C}^*}, \mathsf{st}_{\mathcal{R},i,0}, d'_{i,0}, \mathsf{st}_{\mathcal{R},i,1}, d'_{i,1})$, where $\boldsymbol{\rho}_{\mathcal{C}^*}$ is the final state of the committer after engaging in all four sequential sessions, and $\mathsf{st}_{\mathcal{R},i,0}$, $\mathsf{st}_{\mathcal{R},i,1}$ are receiver states output by the extractor corresponding to the first and third sessions.
- Corresponding to Step 2(b), compute and send $c_i \leftarrow \{0,1\}$.
- Execute Step 2(c) identically to Protocol 1.

– Executes Step 3 of Protocol 1, receiving bits $\{e_i\}_{i\in[\lambda]}$. Fix b^* to be the most frequently ocurring bit in $\{e_i \oplus d'_{i,1-c_i}\}_{i\in[\lambda]}$, and output the final state of $\mathcal{C}^*$, the receiver states $\{\mathsf{st}_{\mathcal{R},i,0}, \mathsf{st}_{\mathcal{R},i,1}\}_{i\in[\lambda]}$, and the extracted bit b^*.

Indistinguishability between the distributions Real and Ideal defined by the above extractor follows by a hybrid argument, and is based on the definition of extractability of the underlying commitment $(\mathsf{Commit}, \mathsf{Decommit})$. In more detail, recall that Real denotes the distribution $(\rho_{\mathcal{C}^*,\mathsf{final}}, b)$ where $\rho_{\mathcal{C}^*,\mathsf{final}}$ denotes the final state of $\mathcal{C}^*$ and b the output of the receiver, and Ideal denotes the final committer state and opened bit after the opening phase of the scheme is run on the output of the extractor.

Note that there are a total of 4λ commitment sessions. For each $i \in [\lambda], j \in [0,3]$, define $\mathsf{Hyb}_{i,j}$ to be the distribution of the committer's state and receiver output when extracting from all commitments in sessions $1,\ldots,i-1$ and extracting from the first j commitments in the i^{th} session, but computing the receiver's output as in the honest protocol.

Claim. There exists a negligible function $\mu(\cdot)$ such that for every $i \in [\lambda], j \in [0,2]$, and every QPT distinguisher $\mathcal{D}$,

$$|\Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i,j}) = 1\big] - \Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i,j+1}) = 1\big]| = \mu(\lambda),$$

and for every $i \in [\lambda]$ and every QPT distinguisher $\mathcal{D}$,

$$|\Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i,3}) = 1\big] - \Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i+1,0}) = 1\big]| = \mu(\lambda).$$

Proof. Suppose this is not the case, then there exists an adversarial committer $\mathcal{C}^*$, a distinguisher $\mathcal{D}$, a polynomial $p(\cdot)$, and an initial committer state ψ that corresponds to a state just before the beginning of the $(i,j+1)^{th}$ commitment, and where

$$\Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i,j}) = 1\big] - \Pr\big[\mathcal{D}(\mathsf{Hybrid}_{i,j+1}) = 1\big]| \geq \frac{1}{p(\lambda)}.$$

Consider a reduction/adversarial committer $\widetilde{\mathcal{C}}$ that obtains initial state ψ, then internally runs $\mathcal{C}^*$, forwarding all messages between an external receiver and $\mathcal{C}^*$ for the $(i,j+1)^{th}$ commitment. It then begins the opening phase, running $\mathcal{C}^*$ internally and forwarding the opening of the $(i,j+1)^{th}$ commitment (if it is executed) to an external receiver. Finally, it outputs the final state of the committer, and b is output by the external receiver. The claim being false directly implies that $\widetilde{\mathcal{C}}$ contradicts extractability of the bit commitment. □

Now for every commitment strategy, every $i \in [\lambda]$, the probability that $d'_{i,1-c_i}$ is not equal to the other bit committed in its pair, and yet the receiver does not abort in Step 2(c) in the i^{th} sequential repetition, is $\leq \frac{1}{2} + \text{negl}(\lambda)$. Then with probability $1 - \text{negl}(\lambda)$, the same also holds for the extracted bits. Thus, by the correctness of the extractor, this implies that the probability that an adversarial committer opens to $1 - b^*$ is at most $1/2^{\lambda/2} + \text{negl}(\lambda) = \text{negl}(\lambda)$. This implies that $\mathsf{Hybrid}_{\lambda,2}$ is indistinguishable from the Ideal distribution defined by the extractor defined above, since the only difference lies in the computation of the receiver's output b^*. Since Real is indistinguishable from $\mathsf{Hybrid}_{1,0}$, this completes the proof.

5 Quantum Extractable Commitments

We construct extractable commitments by making use of the following building blocks.

- We let $(\mathsf{EqCommit}, \mathsf{EqDecommit})$ denote any statistically binding, equivocal quantum commitment scheme. Such a commitment can be obtained by applying the compiler from last section to Naor's commitment scheme [32].
- For a suitable polynomial $k(\cdot)$, let $h : \{0,1\}^{k(\lambda)} \times \{0,1\}^{\lambda^2} \to \{0,1\}$ be a universal hash function that is evaluated on a random seed $s \in \{0,1\}^{k(\lambda)}$ and input $x \in \{0,1\}^{\lambda^2}$.

Our extractable commitment scheme is described formally in Figure 3. We show how to commit to a single bit, though commitment to any arbitrary length string follows by sequential repetition. Correctness of the protocol follows by inspection. In the remainder of this section, we prove the following theorem.

Theorem 2. *Protocol 3 describes a quantum statistically hiding and extractable bit commitment whenever* $(\mathsf{EqCommit}, \mathsf{EqDecommit})$ *is instantiated with any quantum statistically binding and equivocal bit commitment scheme.*

Throughout, we will consider non-uniform adversaries, but for ease of exposition we drop the indexing by λ.

5.1 Extractability

Consider any adversarial committer $\mathcal{C}^*$ with advice $\boldsymbol{\rho}$. The extractor $\mathcal{E}_{\mathcal{C}^*}(\boldsymbol{\rho})$ is constructed as follows.

1. Run the first message algorithm of $\mathcal{C}^*$ on input $\boldsymbol{\rho}$, obtaining message $\boldsymbol{\psi}$.
2. For $i \in [2\lambda^3]$, sequentially execute equivocal commitment sessions with the equivocal simulator $\mathcal{Q}_{R^*,\mathsf{com}}$, where R^* is the part of $\mathcal{C}^*$ that participates as receiver in the i^{th} session. Session i results in output $(\mathbf{z}_i, \mathbf{y}_{\mathsf{com},i})$, where $\mathbf{z}_i$ is stored by the extractor, and $\mathbf{y}_{\mathsf{com},i}$ is the current state of $\mathcal{C}^*$, which is fed as input into the next session.

Protocol 3

Committer $\mathcal{C}$ Input: Bit $b \in \{0,1\}$.

The Protocol: Commit Phase.

1. $\mathcal{C}$ chooses $x \leftarrow \{0,1\}^{2\lambda^3}$, $\theta \leftarrow \{+,\times\}^{2\lambda^3}$ and sends $|x\rangle_\theta$ to $\mathcal{R}$.
2. $\mathcal{R}$ chooses $\widehat{\theta} \leftarrow \{+,\times\}^{2\lambda^3}$ and obtains $\widehat{x} \in \{0,1\}^{2\lambda^3}$ by measuring $|x\rangle_\theta$ in basis $\widehat{\theta}$.
   $\mathcal{R}$ commits to $\widehat{\theta}$ and $\widehat{x}$ position-wise: $\mathcal{R}$ and $\mathcal{C}$ execute sequentially $2\lambda^3$ equivocal commitment sessions with $\mathcal{R}$ as committer and $\mathcal{C}$ as receiver. That is, for each $i \in [2\lambda^3]$, they compute $(\mathbf{x}_{\mathsf{com},i}, \mathbf{y}_{\mathsf{com},i}) \leftarrow \mathsf{EqCommit}\langle \mathcal{R}(\widehat{\theta}_i, \widehat{x}_i), \mathcal{C}\rangle$.
3. $\mathcal{C}$ sends a random test subset $T \subset [2\lambda^3]$ of size λ^3 to $\mathcal{R}$.
4. For every $i \in T$, $\mathcal{R}$ and $\mathcal{C}$ engage in $(\widehat{\theta}_i, \widehat{x}_i) \leftarrow \mathsf{EqDecommit}\langle \mathcal{R}(\mathbf{x}_{\mathsf{com},i}), \mathcal{C}(\mathbf{y}_{\mathsf{com},i})\rangle$, and $\mathcal{C}$ aborts if any commitment fails to open.
5. $\mathcal{C}$ checks that $x_i = \widehat{x}_i$ whenever $\theta_i = \widehat{\theta}_i$. If all tests pass, $\mathcal{C}$ proceeds with the protocol, otherwise, $\mathcal{C}$ aborts.
6. The tested positions are discarded by both parties: $\mathcal{C}$ and $\mathcal{R}$ restrict x and θ, respectively $\widehat{x}$ and $\widehat{\theta}$, to the λ^3 indices $i \in \overline{T}$. $\mathcal{C}$ sends θ to $\mathcal{R}$.
7. $\mathcal{C}$ partitions the remaining λ^3 bits of x into λ different λ^2-bit strings $x^{(1)}, \ldots, x^{(\lambda)}$. For each $\ell \in [\lambda]$, sample a seed $s_\ell \leftarrow \{0,1\}^{k(\lambda)}$ and compute $d_\ell := h(s_\ell, x^{(\ell)})$. Then output $(s_\ell, b \oplus d_\ell)_{\ell \in [\lambda]}$.

The Protocol: Decommit Phase.

1. $\mathcal{C}$ sends b and $(x^{(1)}, \ldots, x^{(\lambda)})$ to $\mathcal{R}$.
2. If either of the following fails, $\mathcal{R}$ rejects and outputs $\bot$. Otherwise, $\mathcal{R}$ accepts and outputs b.
   - Let $\{s_\ell, v_\ell\}_{\ell \in [\lambda]}$ be the message received by $\mathcal{R}$ in step 7. Check that for all $\ell \in [\lambda]$, $v_\ell = b \oplus h(s_\ell, x^{(\ell)})$.
   - For each $j \in [\lambda^3]$ such that $\widehat{\theta}_j = \theta_j$, check that $\widehat{x}_j = x_j$.

Fig. 3. Extractable Commitment.

3. Obtain T from $\mathcal{C}^*$, and sample $\widehat{\theta} \leftarrow \{+,\times\}^{2\lambda^3}$. Let $\boldsymbol{\psi}_i$ denote the i^{th} qubit of $\boldsymbol{\psi}$, and measure the qubits $\boldsymbol{\psi}_i$ for $i \in T$, each in basis $\widehat{\theta}_i$. Let $\{\widehat{x}_i\}_{i \in [T]}$ be the results of the measurements.
4. Let $\mathbf{x}_{\mathsf{com}}$ be the current state of $\mathcal{C}^*$. For each $i \in [T]$, execute $\mathcal{Q}_{R^*,\mathsf{open}}((\widehat{\theta}_i, \widehat{x}_i), \mathbf{z}_i, \mathbf{x}_{\mathsf{com}})$, where R^* is the part of $\mathcal{C}^*$ that participates in the i^{th} opening, and $\mathbf{x}_{\mathsf{com}}$ is updated to be the current state of $\mathcal{C}^*$ after each sequential session.
5. If $\mathcal{C}^*$ aborts at any point, abort and output $\bot$, otherwise continue.
6. Discard tested positions and restrict $\widehat{\theta}$ to the indices in $\overline{T}$. Obtain $\theta \in \{+,\times\}^{\lambda^3}$ from $\mathcal{C}^*$. Measure the qubits $\boldsymbol{\psi}_i$ in basis θ_i to obtain $\widehat{x}_i$ for $i \in \overline{T}$, and then partition $\widehat{x}$ into λ different λ^2-bit strings $\widehat{y}_1, \ldots, \widehat{y}_\lambda$.

7. Obtain $\{s_\ell, v_\ell\}_{\ell\in[\lambda]}$ from $\mathcal{C}^*$. Let b^* be the most frequently occurring bit in $\{h(s_\ell, \widehat{x}^{(\ell)}) \oplus v_\ell\}_{\ell\in[\lambda]}$. Output $(\mathbf{x}_{\mathsf{com}}, \mathbf{y}_{\mathsf{com}}, b^*)$, where $\mathbf{x}_{\mathsf{com}}$ is the resulting state of $\mathcal{C}^*$ and $\mathbf{y}_{\mathsf{com}} = (\theta, \widehat{\theta}, \widehat{x})$.

We now prove that $\mathcal{E}_{\mathcal{C}^*}$ is a secure extractor; for space reasons, a full definition of extractability in the quantum setting is in the supplementary materials (Definition 3.3).

Hyb_1**.** Define distribution Hyb_1 identically to Real (the honest interaction), except that in Step 2, for $i \in [2\lambda^3]$, sequentially execute equivocal commitment sessions using the equivocal simulator $\mathcal{Q}_{R^*,\mathsf{com}}$, as described in the extractor. In Step 4, for every $i \in T$, open the i'th commitment to $(\widehat{\theta}_i, \widehat{x}_i)$ using $\mathcal{Q}_{R^*,\mathsf{open}}$, as described in the extractor.

By the equivocal property of Commit, for any QPT distinguisher $(\mathcal{D}^*, \boldsymbol{\sigma})$, there exists a negligible function $\nu(\cdot)$ such that

$$\Big| \Pr[\mathcal{D}^*(\boldsymbol{\sigma}, \mathsf{Hyb}_1) = 1] - \Pr[\mathcal{D}^*(\boldsymbol{\sigma}, \mathsf{Hyb}_0) = 1] \Big| = \nu(\lambda).$$

Hyb_2**.** This is identical to Hyb_1, except that the verifier measures qubits of $|x\rangle_\theta$ *only after* obtaining a description of the set T, and *only measures* the qubits $i \in [T]$. The output of this experiment is identical to Hyb_1, therefore for any QPT distinguisher $(\mathcal{D}^*, \boldsymbol{\sigma})$,

$$\Pr[\mathcal{D}^*(\boldsymbol{\sigma}, \mathsf{Hyb}_3) = 1] = \Pr[\mathcal{D}^*(\boldsymbol{\sigma}, \mathsf{Hyb}_2) = 1].$$

Moreover, the only difference between Hyb_2 and Ideal is that Ideal outputs FAIL when the message b opened by $\mathcal{C}^*$ is not $\perp$ and differs from the one extracted by $\mathcal{E}_{\mathcal{C}^*}$. Therefore, to derive a contradiction it will suffice to prove that there exists a negligible function $\nu(\cdot)$ such that

$$\Pr[\mathsf{FAIL}|\mathsf{Ideal}] = \nu(\lambda).$$

Consider any sender $\mathcal{C}^*$ that produces a committer state $\mathbf{x}_{\mathsf{com}}$ and then decommits to message b' using strings $(y_1, \ldots, y_\lambda)$ during the decommit phase. Let $T' \subseteq [\lambda]$ denote the set of all indices $\ell \in [\lambda]$ such that the corresponding $x^{(\ell)} \neq v_\ell$, where $\widehat{x}^{(\ell)}$ denotes the values obtained by the extractor in Step 6. Then we have the following claim.

Claim. There exists a negligible function $\nu(\cdot)$ such that

$$\Pr[|T'| > \lambda/2] = \nu(\lambda)$$

where the probability is over the randomness of the extractor.

Proof. For every $\ell \in [\lambda]$, we have that (over the randomness of the extractor):

$$\Pr\left[\mathcal{R}_{\mathsf{open}}(\mathbf{y}_{\mathsf{com}}) \text{ outputs } \perp \text{ in } \langle \mathcal{C}^*_{\mathsf{open}}(\mathbf{x}_{\mathsf{com}}), \mathcal{R}_{\mathsf{open}}(\mathbf{y}_{\mathsf{com}})\rangle \;\Big|\; x^{(\ell)} \neq \widehat{x}^{(\ell)}\right] \geq \frac{1}{2}.$$

Indeed, the receiver will reject if for some position i for which $x^{(\ell)} \neq \widehat{x}^{(\ell)}$, it holds that $\theta_i = \widehat{\theta}_i$. Since $\widehat{\theta}$ was sampled uniformly at random, this will occur for a single i with independent probability 1/2. This implies that $\Pr[|T'| > \lambda/2] \leq \frac{1}{2^{\lambda/2}}$, and the claim follows. □

By construction of $\mathcal{E}_{\mathcal{C}^*}$, $\Pr[\mathsf{FAIL}|\mathsf{Ideal}] < \Pr[|T'| > \lambda/2]$, and therefore it follows that there exists a negligible function $\nu(\cdot)$ such that

$$\Pr[\mathsf{FAIL}|\mathsf{Ideal}] = \nu(\lambda).$$

The proof of that the extractable commitment scheme described in Fig. 3 is statistically hiding follows readily from quantum sampling techniques developed by [8], and is deferred to the supplementary materials.

6 Quantum Oblivious Transfer from Extractable and Equivocal Commitments

6.1 Definitions for Oblivious Transfer with Quantum Communication

An oblivious transfer with quantum communication is a protocol between a quantum interactive sender $\mathcal{S}$ and a quantum interactive receiver $\mathcal{R}$, where the sender $\mathcal{S}$ has input $m_0, m_1 \in \{0,1\}^\lambda$ and the receiver $\mathcal{R}$ has input $b \in \{0,1\}$. After interaction the sender outputs (m_0, m_1) and the receiver outputs (b, m_b).

Let $\mathcal{F}(\cdot,\cdot)$ be the following functionality. $\mathcal{F}(b,\cdot)$ takes as input either (m_0, m_1) or abort from the sender, returns end to the sender, and outputs m_b to the receiver in the non-abort case and $\perp$ in the abort case. $\mathcal{F}(\cdot,(m_0,m_1))$ takes as input either b or abort from the receiver, returns m_b to the receiver, and returns end to the sender in the non-abort case, and returns $\perp$ to the sender in the abort case.

Definition 1. *We let $\langle S(m_0,m_1), R(b)\rangle$ denote an execution of the OT protocol with sender input (m_0,m_1) and receiver input bit b. We denote by $\boldsymbol{\rho}_{\mathsf{out},S^*}\langle S^*(\boldsymbol{\rho}), R(b)\rangle$ and $\mathsf{OUT}_R\langle S^*(\boldsymbol{\rho}), R(b)\rangle$ the final state of a non-uniform malicious sender $S^*(\boldsymbol{\rho})$ and the output of the receiver $R(b)$ at the end of an interaction (leaving the indexing by λ implicit). We denote by $\boldsymbol{\rho}_{\mathsf{out},R^*}\langle S(m_0,m_1), R^*(\boldsymbol{\rho})\rangle$ and $\mathsf{OUT}_S\langle S(m_0,m_1), R^*(\boldsymbol{\rho})\rangle$ the final state of a non-uniform malicious receiver $R^*(\boldsymbol{\rho})$ and the output of the sender $S(m_0,m_1)$ at the end of an interaction. We require OT to satisfy the following security properties:*

– **Receiver Security.** *For every receiver bit $b \in \{0,1\}$, every QPT non-uniform malicious sender $S^*(\boldsymbol{\rho})$, and QPT non-uniform distinguisher $D^*(\boldsymbol{\sigma},\cdot)$, where $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$ may be entangled, there exists a simulator Sim_{S^*} such that the following holds. $\mathsf{Sim}_{S^*}(\boldsymbol{\rho})$ sends inputs (m_0,m_1) or abort to the ideal functionality $\mathcal{F}_{\mathsf{OT}}(b,\cdot)$, whose output to the receiver is denoted by OUT_R.*

$\mathsf{Sim}_{S^*}(\boldsymbol{\rho})$ *also outputs a final state* $\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},S^*}$ *such that*

$$\Big| \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},S^*}, \mathsf{OUT}_R\right)\right) = 1\right]$$
$$- \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{out},S^*}\langle S^*(\boldsymbol{\rho}), R(b)\rangle, \mathsf{OUT}_R\langle S^*(\boldsymbol{\rho}), R(b)\rangle\right)\right) = 1\right]\Big| = \mathrm{negl}(\lambda).$$

- **Sender Security.** *For every pair of sender inputs* (m_0, m_1)*, every QPT non-uniform malicious receiver* $R^*(\boldsymbol{\rho})$*, and QPT non-uniform distinguisher* $D^*(\boldsymbol{\sigma}, \cdot)$*, where* $\boldsymbol{\rho}$ *and* $\boldsymbol{\sigma}$ *may be entangled, there exists a simulator* Sim_{R^*} *such that the following holds.* $\mathsf{Sim}_{R^*}(\boldsymbol{\rho})$ *sends bit* b *or* abort *to the ideal functionality* $\mathcal{F}_{\mathsf{OT}}(m_0, m_1, \cdot)$*, whose output to the sender is denoted by* OUT_S*.* $\mathsf{Sim}_{R^*}(\boldsymbol{\rho})$ *also outputs a final state* $\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},R^*}$ *such that*

$$\Big| \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},R^*}, \mathsf{OUT}_S\right)\right) = 1\right]$$
$$- \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{out},R^*}\langle S(m_0, m_1), R^*(\boldsymbol{\rho})\rangle, \mathsf{OUT}_S\langle S(m_0, m_1), R^*(\boldsymbol{\rho})\rangle\right)\right) = 1\right]\Big| = \mathrm{negl}(\lambda).$$

6.2 Our Construction

We construct simulation-secure quantum oblivious transfer by making use of the following building blocks.

- Let $(\mathsf{EECommit}, \mathsf{EEDecommit})$ denote any quantum bit commitment scheme satisfying extractability and equivocality. Such a commitment scheme may be obtained by applying the compiler from Section 4 to the extractable commitment constructed in Section 5.
- Let $h : \{0,1\}^{k(\lambda)} \times \mathcal{X} \to \{0,1\}^{\lambda}$ be a universal hash with seed length $k(\lambda) = \mathrm{poly}(\lambda)$ and domain $\mathcal{X}$ the set of all binary strings of length *at most* 8λ.

Our QOT protocol is described in Protocol 4, which is essentially the [11] protocol instantiated with our extractable and equivocal commitment scheme.

Theorem 3. *The protocol in Figure 4 is a simulation-secure QOT protocol whenever* $(\mathsf{EECommit}, \mathsf{EEDecommit})$ *is instantiated with a quantum bit commitment satisfying extractability and equivocality.*

We prove that the resulting QOT protocol satisfies standard simulation-based notions of receiver and sender security.

The proof of sender security follows readily from quantum sampling techniques developed by [8], and is deferred to the full version.

6.3 Receiver Security

Consider any adversarial sender S^* with advice $\boldsymbol{\rho}$. The simulator $\mathsf{Sim}_{S^*}(\boldsymbol{\rho})$ is constructed as follows.

Protocol 4

Sender S **Input:** Messages $m_0, m_1 \in \{0,1\}^\lambda \times \{0,1\}^\lambda$
Receiver R **Input:** Bit $b \in \{0,1\}$

The Protocol:

1. S chooses $x \leftarrow \{0,1\}^{16\lambda}$ and $\theta \leftarrow \{+,\times\}^{16\lambda}$ and sends $|x\rangle_\theta$ to R.
2. R chooses $\widehat{\theta} \leftarrow \{+,\times\}^{16\lambda}$ and obtains $\widehat{x} \in \{0,1\}^{16\lambda}$ by measuring $|x\rangle_\theta$ in basis $\widehat{\theta}$. Then, S and R execute 16λ sessions of $\mathsf{EECommit}$ sequentially with R acting as committer and S as receiver. In session i, R commits to the bits $\widehat{\theta}_i, \widehat{x}_i$.
3. S sends a random test subset $T \subset [16\lambda]$ of size 8λ to R.
4. For each $i \in T$, R and S sequentially execute the i'th $\mathsf{EEDecommit}$, after which S receives the opened bits $\widehat{\theta}_i, \widehat{x}_i$.
5. S checks that $x_i = \widehat{x}_i$ whenever $\theta_i = \widehat{\theta}_i$. If all tests pass, S accepts, otherwise, S rejects and aborts.
6. The tested positions are discarded by both parties: S and R restrict x and θ, respectively $\widehat{x}$ and $\widehat{\theta}$, to the 8λ indices $i \in \overline{T}$. S sends θ to R.
7. R partitions the positions of $\overline{T}$ into two parts: the "good" subset $I_b = \{i : \theta_i = \widehat{\theta}_i\}$ and the "bad" subset $I_{1-b} = \{i : \theta_i \neq \widehat{\theta}_i\}$. R sends (I_0, I_1) to S.
8. S samples seeds $s_0, s_1 \leftarrow \{0,1\}^{k(\lambda)}$ and sends $(s_0, h(s_0, x_0) \oplus m_0, s_1, h(s_1, x_1) \oplus m_1)$, where x_0 is x restricted to the set of indices I_0 and x_1 is x restricted to the set of indices I_1.
9. R decrypts s_b using $\widehat{x}_b$, the string $\widehat{x}$ restricted to the set of indices I_b.

Fig. 4. Quantum Oblivious Transfer.

1. Run the first message algorithm of S^* on input $\boldsymbol{\rho}$ to obtain message $\boldsymbol{\psi}$.
2. Execute 16λ sequential sessions of $\mathsf{EECommit}$. In each session, run the equivocator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}$, where $\mathcal{R}^*$ denotes the portion of S^* that participates as receiver in the i^{th} sequential $\mathsf{EECommit}$ session.
3. Obtain test subset T of size 8λ from S^*.
4. For each $i \in T$, sample $\widehat{\theta}_i \leftarrow \{+,\times\}$. Obtain $\widehat{x}_i$ by measuring the i^{th} qubit of $\boldsymbol{\psi}$ in basis $\widehat{\theta}_i$. For each $i \in T$, sequentially execute the equivocal simulator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{open}}$ on input $(\widehat{\theta}_i, \widehat{x}_i)$ and the state obtained from $\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}$.
5. If S^* continues, discard positions indexed by T. Obtain θ_i for $i \in \overline{T}$ from S^*, and compute x_i for $i \in \overline{T}$ by measuring the i^{th} qubit of $\boldsymbol{\psi}$ in basis θ_i.
6. For every $i \in \overline{T}$, sample bit $d_i \leftarrow \{0,1\}$. Partition the set $\overline{T}$ into two subsets (I_0, I_1), where for every $i \in \overline{T}$, place $i \in I_0$ if $d = 0$ and otherwise place $i \in I_1$. Send (I_0, I_1) to S.
7. Obtain (y_0, y_1) from S. Set x_0 to be x restricted to the set of indices I_0 and x_1 to be x restricted to the set of indices I_1. For $b \in \{0,1\}$, parse $y_b = (s_b, t_b)$ and compute $m_b = t_b \oplus h(s_b, x_b)$.

8. If S^* aborts anywhere in the process, send abort to the ideal functionality. Otherwise, send (m_0, m_1) to the ideal functionality. Output the final state of S^*.

Next, we establish receiver security according to Definition 1. Towards a contradiction, suppose there exists a bit $b \in \{0,1\}$, a non-uniform QPT sender $(S^*, \boldsymbol{\rho})$, a non-uniform QPT distinguisher $(D^*, \boldsymbol{\sigma})$, and polynomial $\mathsf{poly}(\cdot)$ s.t.

$$\Big| \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},S^*}, \mathsf{OUT}_R\right)\right) = 1\right] - \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{out},S^*}\langle S^*(\boldsymbol{\rho}), R(b)\rangle, \mathsf{OUT}_R\langle S^*(\boldsymbol{\rho}), R(b)\rangle\right)\right) = 1\right] \Big| \geq \frac{1}{\mathsf{poly}(\lambda)}.$$

Fix any such b, sender $(S^*, \boldsymbol{\rho})$ and distinguisher $(D^*, \boldsymbol{\sigma})$. We derive a contradiction via an intermediate hybrid experiment, defined as follows with respect to bit b and sender $(S^*, \boldsymbol{\rho})$.

Hyb. In this hybrid, we generate the QOT receiver commitments via the equivocal simulator $\mathcal{Q}_{\mathcal{R}^*}$ (where $\mathcal{R}^*$ is derived from the malicious QOT sender S^*), and otherwise follow the honest QOT receiver's algorithm.

1. Run the first message algorithm of S^* on input $\boldsymbol{\rho}$ to obtain message $\boldsymbol{\psi}$.
2. Choose $\widehat{\theta} \leftarrow \{+,\times\}^{16\lambda}$ and obtain $\widehat{x} \in \{0,1\}^{16\lambda}$ by measuring $\boldsymbol{\psi}$ in basis $\widehat{\theta}$. Execute 16λ sequential sessions of $\mathsf{EECommit}$. In each session, run the equivocator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}$, where $\mathcal{R}^*$ denotes the portion of S^* that participates as receiver in the i^{th} sequential $\mathsf{EECommit}$ session.
3. Obtain test subset T of size 8λ from S^*.
4. For each $i \in T$, sequentially execute the equivocal simulator $\mathcal{Q}_{\mathcal{R}^*,\mathsf{open}}$ on input $\widehat{\theta}_i, \widehat{x}_i$ and the state obtained from $\mathcal{Q}_{\mathcal{R}^*,\mathsf{com}}$.
5. If S^* continues, discard positions indexed by T. Obtain θ_i for $i \in \overline{T}$ from S^*.
6. Partition the set $\overline{T}$ into two subsets: the "good" subset $I_b = \{i : \theta_i = \widehat{\theta}_i\}$ and the "bad" subset $I_{1-b} = \{i : \theta_i \neq \widehat{\theta}_i\}$. Send (I_0, I_1) to S.
7. Obtain (y_0, y_1) from S. Set x_b to be $\widehat{x}$ restricted to the set of indices I_b, and compute and set $m_b = t_b \oplus h(s_b, x_b)$. If S^* aborts anywhere in the process, let $\perp$ be the output of the receiver, otherwise let m_b be the output of the receiver.

The output of Hyb is the joint distribution of the final state of S^* and the output of the receiver. Receiver security then follows from the following two claims.

Claim. $\Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{Sim},\mathsf{out},S^*}, \mathsf{OUT}_R\right)\right) = 1\right] \equiv \Pr\left[D^*(\boldsymbol{\sigma}, \mathsf{Hyb}) = 1\right].$

Proof. The only differences in the simulated distribution are (1) that measurements of S^*'s initial message $\boldsymbol{\psi}$ are delayed (which cannot be noticed by S^*), and (2) a syntactic difference in that the ideal functionality is queried to produce the receiver's output. □

Claim. There exists a negligible function $\nu(\cdot)$ such that

$$\Big| \Pr[D^*(\boldsymbol{\sigma}, \mathsf{Hyb}) = 1] - \Pr\left[D^*\left(\boldsymbol{\sigma}, \left(\boldsymbol{\rho}_{\mathsf{out},S^*}\langle S^*(\boldsymbol{\rho}), R(b)\rangle, \mathsf{OUT}_R\langle S^*(\boldsymbol{\rho}), R(b)\rangle\right)\right) = 1\right] \Big| = \nu(\lambda).$$

Proof. The only difference between the two distributions is that in the first, the receiver generates commitments according to the honest commit algorithms of EECommit while in the second, commitments in step 2 are generated via the equivocal simulator $\mathcal{Q}_{\mathcal{R}^*}$ of EECommit. Therefore, this claim follows by the equivocality of (EECommit, EEDecommit) . □

Finally, Theorems 1, Theorem 2, and Theorem 3 give the following.

Corollary 1. *Quantum oblivious transfer (QOT) satisfying Definition 1 can be based on black-box use of statistically binding bit commitments, or on black-box use of quantum-hard one-way functions.*

7 Acknowledgments

The authors are grateful to the Simons Institute programs on *Lattices: Algorithms, Complexity and Cryptography*, and *The Quantum Wave in Computing* for fostering this collaboration. Thanks also to Alex Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan for discussions about similarities and differences with [19]. A.C. is supported by DoE under grant DOD ONR Federal. This material is based on work supported in part by DARPA under Contract No. HR001120C0024 (for DK). 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.

References

1. Aaronson, S.: Quantum copy-protection and quantum money. In: 2009 24th Annual IEEE Conference on Computational Complexity. pp. 229–242. IEEE (2009)
2. Agarwal, A., Bartusek, J., Goyal, V., Khurana, D., Malavolta, G.: Post-quantum multi-party computation. Cryptology ePrint Archive, Report 2020/1395 (2020), https://eprint.iacr.org/2020/1395
3. Barak, B., Mahmoody-Ghidary, M.: Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 374–390. Springer, Heidelberg (Aug 2009)
4. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multi-party quantum computation with (only) a strict honest majority. In: 47th FOCS. pp. 249–260. IEEE Computer Society Press (Oct 2006)
5. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing. pp. 175–179 (1984)
6. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, M.H.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 351–366. Springer, Heidelberg (Aug 1992)

7. Bitansky, N., Shmueli, O.: Post-quantum zero knowledge in constant rounds. In: Makarychev, K., Makarychev, Y., Tulsiani, M., Kamath, G., Chuzhoy, J. (eds.) Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020. pp. 269–279. ACM (2020)
8. Bouman, N.J., Fehr, S.: Sampling in a quantum population, and applications. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 724–741. Springer, Heidelberg (Aug 2010)
9. Brassard, G., Crépeau, C., Jozsa, R., Langlois, D.: A quantum bit commitment scheme provably unbreakable by both parties. In: 34th FOCS. pp. 362–371. IEEE Computer Society Press (Nov 1993)
10. Crépeau, C., Gottesman, D., Smith, A.: Secure multi-party quantum computation. In: 34th ACM STOC. pp. 643–652. ACM Press (May 2002)
11. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: 29th FOCS. pp. 42–52. IEEE Computer Society Press (Oct 1988)
12. Crépeau, C., van de Graaf, J., Tapp, A.: Committed oblivious transfer and private multi-party computation. In: Coppersmith, D. (ed.) CRYPTO'95. LNCS, vol. 963, pp. 110–123. Springer, Heidelberg (Aug 1995)
13. Damgård, I., Fehr, S., Lunemann, C., Salvail, L., Schaffner, C.: Improving the security of quantum protocols via commit-and-open. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 408–427. Springer, Heidelberg (Aug 2009)
14. Damgård, I., Ishai, Y.: Constant-round multiparty computation using a black-box pseudorandom generator. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 378–394. Springer, Heidelberg (Aug 2005)
15. 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.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 729–758. Springer, Heidelberg (May 2020)
16. 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 (Aug 2010)
17. Dupuis, F., Nielsen, J.B., Salvail, L.: Actively secure two-party evaluation of any quantum operation. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012. Proceedings. LNCS, vol. 7417, pp. 794–811. Springer (2012)
18. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 171–185. Springer, Heidelberg (Aug 1987)
19. Grilo, A.B., Lin, H., Song, F., Vaikuntanathan, V.: Oblivious transfer is in miniqcrypt. CoRR **abs/2011.14980** (2020)
20. 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 (Aug 2011)
21. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
22. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions (extended abstracts). In: 21st ACM STOC. pp. 12–24. ACM Press (May 1989)
23. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography (extended abstract). In: 30th FOCS. pp. 230–235. IEEE Computer Society Press (Oct / Nov 1989)

24. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 8–26. Springer, Heidelberg (Aug 1990)
25. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (Aug 2008)
26. Kilian, J.: Founding cryptography on oblivious transfer. In: 20th ACM STOC. pp. 20–31. ACM Press (May 1988)
27. Lo, H.K., Chau, H.F.: Is quantum bit commitment really possible? Physical Review Letters **78**(17), 3410 (1997)
28. Luby, M., Rackoff, C.: Pseudo-random permutation generators and cryptographic composition. In: 18th ACM STOC. pp. 356–363. ACM Press (May 1986)
29. Mahmoody, M., Maji, H.K., Prabhakaran, M.: On the power of public-key encryption in secure computation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 240–264. Springer, Heidelberg (Feb 2014)
30. Mayers, D.: Unconditionally secure quantum bit commitment is impossible. Physical review letters **78**(17), 3414 (1997)
31. Mayers, D., Salvail, L.: Quantum oblivious transfer is secure against all individual measurements. In: Proceedings Workshop on Physics and Computation. PhysComp'94. pp. 69–77. IEEE (1994)
32. Naor, M.: Bit commitment using pseudorandomness. Journal of Cryptology **4**(2), 151–158 (Jan 1991)
33. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (Aug 2007)
34. Unruh, D.: Quantum position verification in the random oracle model. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 1–18. Springer, Heidelberg (Aug 2014)
35. Watrous, J.: Zero-knowledge against quantum attacks. In: Kleinberg, J.M. (ed.) 38th ACM STOC. pp. 296–305. ACM Press (May 2006)
36. Wolf, S., Wullschleger, J.: Oblivious transfer is symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (May / Jun 2006)
37. Yao, A.C.C.: Security of quantum protocols against coherent measurements. In: 27th ACM STOC. pp. 67–75. ACM Press (May / Jun 1995)
38. Zhandry, M.: How to construct quantum random functions. In: 53rd FOCS. pp. 679–687. IEEE Computer Society Press (Oct 2012)