Post-Quantum Simulatable Extraction with Minimal Assumptions: Black-Box and Constant-Round

Nai-Hui Chia[1], Kai-Min Chung[2*], Xiao Liang[3†], and Takashi Yamakawa[4]

[1] Indiana University Bloomington, IN, USA
naichia@iu.edu
[2] Academia Sinica, Taipei, Taiwan
kmchung@iis.sinica.edu.tw
[3] Stony Brook University, NY, USA
liang1@cs.stonybrook.edu
[4] NTT Social Informatics Laboratories, Tokyo, Japan
takashi.yamakawa.ga@hco.ntt.co.jp

Abstract. From the minimal assumption of post-quantum semi-honest oblivious transfers, we build the first ε-*simulatable* two-party computation (2PC) against quantum polynomial-time (QPT) adversaries that is both constant-round and black-box (for both the construction and security reduction). A recent work by Chia, Chung, Liu, and Yamakawa (FOCS'21) shows that post-quantum 2PC with standard simulation-based security is impossible in constant rounds, unless either $\mathbf{NP} \subseteq \mathbf{BQP}$ or relying on non-black-box simulation. The ε-simulatability we target is a relaxation of the standard simulation-based security that allows for an arbitrarily small noticeable simulation error ε. Moreover, when quantum communication is allowed, we can further weaken the assumption to post-quantum secure one-way functions (PQ-OWFs), while maintaining the constant-round and black-box property.

Our techniques also yield the following set of *constant-round and black-box* two-party protocols secure against QPT adversaries, only assuming black-box access to PQ-OWFs:

- extractable commitments for which the extractor is also an ε-simulator;
- ε-zero-knowledge commit-and-prove whose commit stage is extractable with ε-simulation;
- ε-simulatable coin-flipping;
- ε-zero-knowledge arguments of knowledge for **NP** for which the knowledge extractor is also an ε-simulator;
- ε-zero-knowledge arguments for **QMA**.

At the heart of the above results is a black-box extraction lemma showing how to efficiently extract secrets from QPT adversaries while disturbing their quantum state in a controllable manner, i.e., achieving ε-simulatability of the post-extraction state of the adversary.

Keywords: Simulation · Extraction · Post-Quantum

* Kai-Min Chung is supported in part by Ministry of Science and Technology, Taiwan, under Grant no. MOST 109-2223-E-001-001-MY3, the 2021 Academia Sinica Investigator Award (AS-IA-110-M02), and the Air Force Office of Scientific Research under award number FA2386-20-1-4066.

† Xiao Liang is supported in part by Omkant Pandey's DARPA SIEVE Award HR00112020026 and NSF grants 1907908 and 2028920. 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, DARPA, or NSF.

Table of Contents

1 Introduction

Extractability is an important concept in cryptography. A typical example is extractable commitments, which enable an extractor to extract a committed message from a malicious committer. Extractable commitments have played a central role in several major cryptographic tasks, including (but not limited to) secure two-party and multi-party computation (e.g., [CDMW09, PW09, Goy11, GLP+15]), zero-knowledge (ZK) protocols (e.g., [Ros04, Lin13]), concurrent zero-knowledge protocols (e.g., [PRS02, MOSV06]), non-malleable commitments (e.g., [GLOV12, Kiy14]) etc. Recently, two concurrent works by Grilo, Lin, Song, and Vaikuntanathan [GLSV21] and Bartusek, Coladangelo, Khurana, and Ma [BCKM21] (based on earlier works [CK90, BBCS92, DFL+09, BF10]) demonstrate new applications of extractable commitments in quantum cryptography. They show that quantumly secure extractable commitments are sufficient for constructing maliciously secure quantum oblivious transfers (OTs), which can be compiled into general-purpose quantum MPC [IPS08, DGJ+20].[5]

As noted in [GLSV21], it is surprisingly non-trivial to construct quantumly secure extractable commitments. The reason is that quantum extractability requires an extractor to extract the committed message *while simulating the committer's post-execution state.* However, known rewinding-based classical extraction techniques are not directly applicable as it is unclear if they could provide any simulation guarantee when used against quantum adversaries. To address this issue, recent works [GLSV21, BCKM21] propose new polynomial-round *quantum* constructions of quantumly secure extractable commitments from post-quantum one-way functions (PQ-OWFs), which are functions efficiently computable in the classical sense but one-way against quantum polynomial-time (QPT) adversaries. Relying on assumptions stronger than PQ-OWFs, *classical* constructions of quantumly secure extractable commitments (which we call *post-quantum* extractable commitments) are known [BS20, AL20, BLS21, HSS11, LN11]. However, those constructions require (at least) the existence of OTs.

Moreover, all existing post-quantum extractable commitments make *non-black-box* use of their building-block primitives. This is not ideal as *black-box* constructions are often preferred over non-black-box ones. A black-box construction only depends on the input/output behavior of its building-block cryptographic primitive(s). In particular, such a construction is independent of the specific implementation or code of the building-block primitive. Black-box constructions enjoy certain advantages. For example, they remain valid even if the building-block primitive/oracle is based on a *physical* object such as a noisy channel or tamper-proof hardware [Wyn75, CK88, GLM+04]. Also, since the efficiency of black-box constructions does not depend on the implementation details of the primitive, their efficiency can be theoretically independent of the code of lower-level primitives. Indeed, it has been an important theme to obtain black-box constructions for major cryptographic objects, e.g., [Kil88, DI05, IKLP06, IKOS07, Hai08, IPS08, PW09, Goy11, GLOV12, LP12, Kiy14, GOSV14, GGMP16, HV16, GKP18, KOS18, CLP20, GLPV20, GKLW21, LP21, CCY21].

In the classical setting, it is well-known that constant-round extractable commitments can be obtained assuming only black-box access to OWFs [PW09, DDN00, PRS02, Ros04].[6] Therefore, it is natural to ask the following analog question in the quantum setting: Is it possible to construct constant-round post-quantum extractable commitments assuming only black-box access to PQ-OWFs? We remark that this question is open even if we do not require the scheme to be constant-round or black-box.

[5] They actually rely on extractable and *equivocal* commitments. However, since equivocality can be added easily, extractable commitments are the essential building block.

[6] The term "black-box" here refers to both black-box construction and black-box extraction.

The Black-Box Extraction Barrier. We observe that the recent lower bound on black-box post-quantum ZK [CCLY21] suggests a negative answer to the above question. Namely, if we have constant-round post-quantum extractable commitment with black-box extraction, then we can construct constant-round post-quantum ZK arguments for **NP** with black-box simulation based on standard techniques (see Appx. A for details). However, [CCLY21] showed that such a ZK argument cannot exist unless $\mathbf{NP} \subseteq \mathbf{BQP}$, which seems unlikely.[7]

ε-Simulation Security. On the other hand, another recent work [CCY21] showed that we can bypass the impossibility result by relaxing the requirement of ZK to the so-called ε-ZK [DNS04, BKP18, FGJ18]. The standard ZK property requires a simulator to simulate the verifier's view in a way that no distinguisher can distinguish it from the real one with non-negligible advantage. In contrast, the ε-ZK property only requires the existence of a simulator such that for any noticeable $\varepsilon(\lambda)$, the simulated view can be distinguished from the real one with advantage at most ε. As explained in [CCY21], ε-ZK is still useful in several applications of ZK. The results in [CCY21] suggest the possibility of post-quantum extractable commitments if we relax the simulation requirement on the extractor to a similar ε-close[8] version. We will refer to this weakened notion as *extractability with ε-simulation*.[9] It seems natural to hope that the techniques in [CCY21] could be used in the context of extractable commitments. Indeed, by plugging the ZK argument from [CCY21] into the OT-based construction [BS20, BLS21, HSS11, LN11], we can obtain a non-black-box construction of constant-round post-quantum extractable commitments with ε-simulation, assuming constant-round post-quantum OTs. However, if we focus on *black-box constructions* from the *minimal assumption* of PQ-OWFs, it is unclear if the techniques in [CCY21] would help. Therefore, we ask the following question:

> **Question 1:** *Is it possible to have constant-round post-quantum extractable commitments with ε-simulation, assuming only black-box access to PQ-OWFs?*

In the more general context of 2PC and MPC, the implication of [CCLY21] is that to obtain constant-round constructions with post-quantum security, we have to either

1. rely on non-black-box simulation, *or*
2. aim for a relaxed security notion (e.g., ε-close simulation security).

The first approach was taken in [ABG+21a] (based on [BS20]), leading to a constant-round post-quantum MPC protocol with non-black-box simulation. On the other hand, the second approach has not been explored in the existing literature of post-qauntum 2PC or MPC (except for the special case of ZK as in [CCY21]). It is possible to construct constant-round post-quantum 2PC with ε-close simulation by combining constant-round post-quantum semi-honest OTs and the constant-round post-quantum ε-ZK in [CCY21]. However, the naive approach will lead to a non-black-box construction. In contrast, in the classical setting, constant-round *black-box constructions* of 2PC [PW09] and MPC [CDMW09, Goy11] are known from the minimal assumption of constant-round semi-honest OT. The above discussion suggests that one has to relax the security requirement when considering the post-quantum counterparts of these tasks. We will refer to 2PC and MPC with ε-close simulation as ε-2PC and ε-MPC respectively. Then, an interesting question is:

> **Question 2:** *Do there exist constant-round black-box post-quantum ε-2PC and ε-MPC, assuming only constant-round semi-honest OTs secure against QPT adversaries?*

[7] A concurrent work by Lombardi, Ma, and Spooner [LMS21] showed that the impossibility of [CCLY21] can be avoided if we consider a stronger computational model for simulators. We provide more discussion in Sec. 1.3.

[8] Throughout this paper, "ε-close" means that the adversary's distinguishing advantage is at most ε.

[9] In the main body, we call it *strong* extractability with ε-simulation since we also define a weaker variant of that.

Table 1. Comparison of Quantumly Secure Extractable Commitment.

Reference	#Round	Cla. Const.	BB Const.	BB Ext.	Siml. Err.	Assumption
[GLSV21]	$\mathsf{poly}(\lambda)$			✓	negl	OWF
[BCKM21]	$\mathsf{poly}(\lambda)$		✓	✓	negl	OWF
[BS20]	$O(1)$	✓			negl	QFHE+QLWE
folklore[a]	$\mathsf{poly}(\lambda)$	✓		✓	negl	OT
folklore+[CCY21]	$O(1)$	✓		✓	ε	$O(1)$-round OT
Ours	$O(1)$	✓	✓	✓	ε	OWF

The "Cla. Const.", "BB Const.", and "BB Ext." columns indicate if the scheme relies on classical constructions, black-box constructions, and extraction, respectively. In the "Siml. Err." column, negl and ε mean that the construction achieves the standard quantum extractability and quantum extractability with ε-simulation, respectively. In "Assumption" column, QFHE and QLWE means quantum fully homomorphic encryption and the quantum hardness of learning with errors, respectively.

[a] As noted in [BLS21], the construction is implicit in [BS20, HSS11, LN11].

1.1 Our Results

We answer **Question 1** affirmatively and address **Question 2** partially, showing a positive answer only for the two-party case. We first construct constant-round black-box post-quantum extractable commitments with ε-simulation from PQ-OWFs. See Table 1 for comparisons among quantumly secure extractable commitments. Such commitments imply new constant-round and black-box protocols for general-purpose 2PC secure against QPT adversaries. In particular, we get

- post-quantum ε-2PC from semi-honest OTs, and
- post-quantum ε-2PC from PQ-OWFs, assuming that quantum communication is possible. (Henceforth, we will use OWFs to denote PQ-OWFs.)

As an intermediate tool to achieve the above results, we construct a constant-round post-quantum ε-ZK commit-and-prove, assuming only black-box access to OWFs. Black-box zero-knowledge commit-and-prove [IKOS07, GLOV12, GOSV14, HV18, KOS18, Kiy20] is a well-studied primitive in classical cryptography. It enables a prover to commit to some message and later to prove in zero-knowledge that the committed message satisfies a given predicate in a *black-box* manner. In addition to being secure in the post-quantum setting, our construction enjoys the extra property that the commit stage is extractable (albeit with only ε-simulation of the adversary's post-extraction state). Such a constant-round ε-simulatable ExtCom-and-Prove protocol implies the following set of two-party protocols:

- constant-round black-box post-quantum coin-flipping with ε-simulation,
- constant-round black-box post-quantum ε-ZK arguments of knowledge for **NP** with ε-simulating knowledge extractor, and
- constant-round black-box ε-ZK arguments for **QMA**.

In the following, we provide more discussion about them.

Coin-Flipping. Coin-flipping is a two-party protocol used to generate a uniformly random string that cannot be biased by either of parties (w.r.t. the standard simulation-based security). In the classical setting, constant-round black-box constructions from OWFs are known [PW09]. On the other hand, known post-quantum constructions are based on stronger assumptions (like QLWE)

than OWFs, and require either polynomial rounds [LN11] or non-black-box simulation [ABG+21a]. Our construction can be understood as the post-quantum counterpart of the classical construction by Pass and Wee [PW09], albeit with ε-simulation.

Arguments of Knowledge with Simulating Extractor. Arguments of knowledge intuitively require an extractor to extract a witness from any efficient malicious prover whenever it passes the verification. In the classical setting, constant-round black-box constructions from OWFs are known [PW09]. In the post-quantum setting, there are two existing notions of arguments of knowledge depending on whether we require the extractor to simulate the prover's post-execution state or not. For the "without-simulation" version, Unruh [Unr12] gave a polynomial-round black-box construction from OWFs.[10] For the "with-simulation" version, all existing constructions require both polynomial rounds *and* assumptions stronger than OWFs (like QLWE) [HSS11, LN11, ACP21].[11] Our construction improves both the round complexity and the required assumption, at the cost of weakening ZK and extractability to their ε-simulation variants. On the other hand, we note that the construction in [ACP21] achieves *proofs of knowledge*, while ours only achieves *arguments of knowledge*. We also note that even without knowledge extractability, our construction improves the construction in [CCY21, Section 6], which is a *non-black-box construction* of constant-round ε-ZK arguments for **NP** from OWFs.

ZK Arguments for QMA. QMA is a quantum analog of **NP**. Known constructions of ZK proofs or arguments for **QMA** rely on either polynomial-round communication [BJSW20, BG20, BY20] or non-black-box simulation [BS20]. If we relax the ZK requirement to ε-ZK, constant-round black-box ε-ZK *proofs* were already constructed in [CCY21]; but that construction needs to assume collapsing hash functions, which are stronger than OWFs. Our construction improves the assumption to the existence of OWFs at the cost of weakening the soundness to the computational one (i.e., an argument system).

1.2 Discussion

Minimality of Assumptions. We can show that OWFs and semi-honest OTs are the minimal assumptions for post-quantum extractable commitments with ε-simulation and ε-2PC, respectively. The former is straightforward (since any computationally-hiding and statistically-binding commitments without extractability already imply OWFs), but the latter needs more explanations. First, we note that ε-2PC trivially implies ε-simulatable semi-honest OTs because the latter is a special case of the former. Next, we remark that ε-simulatable semi-honest security implies the standard semi-honest indistinguishability-based security (with negligible distinguishing advantage). This is a special case of a folklore that ε-simulation security suffices for indistinguishability-based applications. The reason is that we can set the simulation error ε after an adversary's distinguishing advantage δ is fixed so that $\varepsilon \ll \delta$, because the simulator only appears in the security proof for the indistinguishability-based security. Finally, we remark that semi-honest indistinguishability-based security implies semi-honest simulation-based security. In summary, ε-2PC implies standard semi-honest OTs.

We also remark that, as observed in [BCKM21, GLSV21], it is unclear if OWFs are necessary for *quantum constructions* of 2PC/MPC, and it may be possible to construct them based on a weaker assumption.

[10] Though Unruh originally assumes *injective* OWFs, [CCY21] pointed out that any OWF suffices.

[11] Though not claimed explicitly, it seems also possible to obtain constant-round construction with non-black-box simulation from QLWE and QFHE based on [BS20].

Other Potential Applications. A recent work by Bitansky, Lin, and Shmueli [BLS21] gave a generic construction of post-quantum non-malleable commitments from post-quantum extractable commitments. We believe that our ε-simulatable extractable commitments can be used in their construction as a building block. It is reasonable to expect that we can prove the standard non-malleability as defined in [BLS21] even though we start from ε-simulatable extractability; This is because non-malleability is an indistinguishability-based security and ε-simulation usually suffices for indistinguishability-based applications as mentioned in the previous paragraph. If the above idea works, we will get $\log^\star(\lambda)$-round post-quantum non-malleable commitments solely from OWFs. [BLS21] obtains their $\log^\star(\lambda)$-round protocol based on much stronger assumptions of QFHE and QLWE; they also presents a polynomial-round classical (resp. quantum) protocol based on OTs (resp. OWFs). While our intuition above is plausible, a formal proof of it is out of scope of this work, and thus is left for future work.

1.3 Concurrent Work

A concurrent work by Lombardi, Ma, and Spooner [LMS21] observed that the impossibility of [CCLY21] implicitly assumed a computational model for simulators, which they call measured-runtime expected quantum polynomial time (EQPT_m), and showed that the impossibility can be circumvented if we consider a stronger model called coherent-runtime expected quantum polynomial time (EQPT_c). Roughly speaking, the difference between EQPT_m and EQPT_c is that the latter allows the simulator to coherently run multiple computations with different runtimes so that they can interfere with each other. (See [LMS21] for more details.) Then, they construct constant-round post-quantum zero-knowledge proofs (or arguments) and extractable commitments with EQPT_c simulators and extractors. Though ZK with EQPT_c simulators is weaker than the standard ZK with polynomial-time simulators, they show that it implies ε-ZK. Therefore, their notion of ZK with EQPT_c simulation lies between the standard ZK and ε-ZK. Their formalization of EQPT_c simulation is very interesting as this enables us to reduce the simulation error to be negligible, which makes the state of affairs be similar to the classical case.

On the other hand, we believe that the gap between ZK with EQPT_c simulation and ε-ZK is not too large from the perspective of (theoretical) applications: The definition of EQPT_c models a simulator as a quantum circuit of a superpolynomial sizes with a certain property, and in particular, if we need a polynomial-time simulator, we must truncate the simulation after running polynomially many steps. However, in that case, there occurs a noticeable simulation error, which is similar to ε-ZK. For this reason, we do not find any application for which ZK with EQPT_c simulation suffices but ε-ZK does not. We make a similar observation on extractability with EQPT_c extraction and extractability with ε-close simulation as well.

Below, we give a comparison of results of [LMS21] and our work. Among many others, [LMS21] constructed constant-round post-quantum extractable commitments with EQPT_c extraction assuming super-polynomially secure OWFs or polynomially secure collapse-binding commitments. Though they achieve a stronger notion of extractability than ours as explained above, they rely on non-black-box constructions and stronger assumptions than the polynomial hardness of OWFs. On the other hand, the main focus of this work is black-box constructions from the minimal assumption of the polynomial hardness of OWFs (or OTs for 2PC). There are similar advantages and disadvantages for all constructions given in [LMS21]. Thus, the results of [LMS21] are incomparable to ours.

A natural question that arises from the above comparison is if we can obtain protocols that take advantages of both works. That is, can we construct constant-round black-box post-quantum extractable commitments (resp. 2PC) with EQPT_c extraction (resp. simulation) and negligible

simulation errors assuming only the minimal assumption of polynomially secure OWFs (resp. OTs)? This might be achieved by combining the techniques of this paper and [LMS21]. We leave it as an interesting future work.

2 Technical Overview

We give technical overview for our results on extractable commitments, ExtCom-and-Prove, and 2PC. The other applications claimed before follows from our ExtCom-and-Prove protocol via rather standard techniques. Therefore, we refer the reader to Sec. 6 for corresponding constructions.

2.1 Extractable Commitment with ε-Simulation

The main technical tool for constructing extractable commitments with ε-simulation is a generalization of the recent extract-and-simualate technique of [CCY21].

Extract-and-Simulation Lemma in [CCY21]. We briefly recall the extract-and-simulate lemma shown in [CCY21, Lemma 4.2].[12] At a high level, that lemma can be interpreted as follows.[13] Let $\mathcal{A}$ be a quantum algorithm with an initial state ρ. Suppose that $\mathcal{A}$ outputs some unique classical string s^* or otherwise outputs a failure symbol Fail. Then, there exists a simulation-extractor $\mathcal{SE}$ such that for any noticeable function ε (on the security parameter), the following two experiments are ε-close:

$\underline{\mathsf{Exp}_{\mathsf{real}}}$	$\underline{\mathsf{Exp}_{\mathsf{ext}}}$
	$(s_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \leftarrow \mathcal{SE}^{\mathcal{A}(\rho)}(1^{\varepsilon^{-1}})$
Run $\mathcal{A}(\rho)$,	Run $\mathcal{A}(\rho_{\mathsf{Ext}})$,
If $\mathcal{A}$ outputs Fail,	*If* $\mathcal{A}$ outputs Fail $\vee$ $s_{\mathsf{Ext}} \neq s^*$,
Output Fail	*Output* Fail
Else output $\mathcal{A}$'s final state.	*Else output* $\mathcal{A}$'s final state.

Generalizing the Lemma. Note that their lemma will enable us to extract s^* from $\mathcal{A}$ *only if* $\mathcal{A}$ *reveals the value* s^* *at the end*. As shown in [CCY21], this already suffices for the constant-round ZK proof by Goldreich and Kahan [GK96], where the verifier first commits to the challenge and opens it (i.e., "reveals it at the end") later. However, this does not seem to help obtain extractable commitments, because the committed message is not revealed *at the end the commit stage* (i.e., before decommitment happens); but the definition of extractable commitments does require extraction before decommitment happens.

To deal with this issue, we generalize the [CCY21] lemma as follows. Let $\mathcal{A}$ be a quantum algorithm that on an initial state ρ, outputs a classical symbol Succ or Fail. Moreover, suppose that there are a unique classical string s^* and a "simulation-less extractor" $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$ that outputs s^*, or otherwise Fail. Also, suppose that

$$\Pr\left[\mathsf{Ext}_{\mathsf{Sim\text{-}less}}^{\mathcal{A}(\rho)} = s^*\right] \geq (\Pr[\mathcal{A}(\rho) = \mathsf{Succ}])^c - \mathsf{negl}(\lambda) \tag{1}$$

for some constant c. Our generalized lemma says that the ε-closeness between $\mathsf{Exp}_{\mathsf{real}}$ and $\mathsf{Exp}_{\mathsf{ext}}$ holds in this setting as well.

One can think of $\mathcal{A}$ as a joint execution of a malicious committer and honest receiver where it outputs Succ if and only if the receiver accepts. In this setting, one can understand the above lemma

[12] In [CCY21], the lemma was called "extraction lemma". Here, we add "simulation" to emphasize that the extractor not only extracts but also simulates the adversary's state.

[13] There are two versions of their lemma: the statistically-binding case and the strong collapse-binding case. The abstraction given here is a generalization of the statistically-binding case.

as a lifting lemma from "simulation-less extractor" to "ε-simulation extractor" in the setting where the extracted string is unique. In the main body, we present the lemma in a more specific form (Lem. 4), where it is integrated with Watrous' rewinding lemma [Wat09] and Unruh's rewinding lemma [Unr12], because that is more convenient for our purpose. We will overview the intuition behind the above generalized lemma toward the end of this subsection.

Weakly Extractable Commitment. Next, we explain how to construct post-quantum extractable commitments using our extract-and-simulate lemma. We go through the following two steps:

1. Construct a commitment scheme $\mathsf{wExtCom}$ that satisfies a weak version of post-quantum extractability with ε-simulation.
2. Upgrade $\mathsf{wExtCom}$ into a scheme ExtCom with full-fledged post-quantum extractability with ε-simulation (which we call *strong* extractability with ε-simulation to distinguish it from the weak one).

We first explain Step 1, the construction of $\mathsf{wExtCom}$. Actually, our construction of $\mathsf{wExtCom}$ is exactly the same as the classical extractable commitments from OWFs given in [PW09], which are in turn based on earlier works [DDN00, PRS02, Ros04]. Let Com be a computationally-hiding and statistically-binding commitment scheme (say, Naor's commitment [Nao91]). Then, the commitment scheme $\mathsf{wExtCom}$ works as follows.

Commit Stage:

1. To commit to a message m, the committer C generates $k = \omega(\log \lambda)$ pairs of 2-out-of-2 additive secret shares $\{(v_i^0, v_i^1)\}_{i=1}^k$, i.e., they are uniformly chosen conditioned on that $v_i^0 \oplus v_i^1 = m$ for each $i \in [k]$. Then, C commits independently to each v_i^b ($b \in \{0,1\}$) in parallel by using Com. We denote these commitments by $\{(\mathsf{com}_i^0, \mathsf{com}_i^1)\}_{i=1}^k$.
2. R randomly chooses $\mathbf{c} = (c_1, ..., c_k) \leftarrow \{0,1\}^k$ and sends it to C.
3. C decommits $\{\mathsf{com}_i^{c_i}\}_{i=1}^k$ to $\{v_i^{c_i}\}_{i=1}^k$, and R checks that the openings are valid.

Decommit Stage:

1. C sends m and opens all the remaining commitments; R checks that all openings are valid *and* $v_i^0 \oplus v_i^1 = m$ for all $i \in [k]$.

Suppose that a malicious committer C^* generates commitments $\{(\mathsf{com}_i^0, \mathsf{com}_i^1)\}_{i=1}^k$ in Step 1, and let ρ be its internal state at this point. Then, we consider $\mathcal{A}(\rho)$ that works as follows:

– Choose $\mathbf{c} = (c_1, ..., c_k) \leftarrow \{0,1\}^k$ at random.

– Send $\mathbf{c}$ to C^* and simulate Step 3 of C^* in the commit stage to get $\{v_i^{c_i}\}_{i=1}^k$ and the corresponding decommitment information.

– If all the openings are valid, output Succ; otherwise output Fail.

To use our extract-and-simulate lemma, we need to construct a simulation-less extractor $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$ satisfying Inequality (1). A natural idea is to use Unruh's rewinding lemma [Unr12]. His lemma directly implies that if $\mathcal{A}$ returns Succ with probability δ, then we can obtain valid $\{v_i^{c_i}\}_{i=1}^k$ and $\{v_i^{c'_i}\}_{i=1}^k$ for two uniformly random challenges, $\mathbf{c} = (c_1, \ldots, c_k)$ and $\mathbf{c}' = (c'_1, \ldots, c'_k)$, with probability at least δ^3. In that case, unless $\mathbf{c} = \mathbf{c}'$ (which happens with negligible probability), we can "extract" $m = v_i^0 \oplus v_i^1$ from position $i \in [k]$ that satisfies $c_i \neq c'_i$. However, such an "extractor" does not satisfy the assumption for our generalized extract-and-simulate lemma in general, because $v_i^0 \oplus v_i^1$ may be different for each $i \in [k]$.

Therefore, to satisfy this requirement, we have to introduce an additional assumption that $\{(\mathsf{com}_i^0, \mathsf{com}_i^1)\}_{i=1}^k$ is *consistent*, i.e., if we denote the corresponding committed messages as $\{(v_i^0, v_i^1)\}_{i=1}^k$, then there exists a unique m such that $v_i^0 \oplus v_i^1 = m$ for all $i \in [k]$.[14] With this assumption, we can apply our generalized extract-and-simulate lemma. It enables us to extract the committed message *and* simultaneously ε-simulate C^*'s state, conditioned on that the receiver accepts in the commit stage. The case where the receiver rejects can be easily handled using Watrous' rewinding lemma [Wat09] as we will explain later. As a result, we get an ε-simulating extractor that works well conditioned on that the commitments generated in Step 1 are consistent. We will refer to such a weak notion of simulation-extractability as *weak extractability with ε-simulation* (see Def. 13 for the formal definition).

Moreover, since Unruh's rewinding lemma naturally gives a simulation-less extractor in the parallel setting (where C^* interacts with many copies of R in parallel), we can prove the parallel version of the weak extractability with ε-simulation similarly. More generally, we prove that $\mathsf{wExtCom}$ satisfies a further generalized notion of extractability which we call the *special parallel weak extractability with ε-simulation* (see Def. 16 for the formal definition). Roughly speaking, it requires an ε-simulating extractor to work in n-parallel execution as long as the commitments in some subset of $[n]$ are consistent and the committed messages in those sessions determine a unique value. This parallel extractability will play an important role in the weak-to-strong compiler which we discuss next.

Weak-to-Strong Compiler. The reason why we cannot directly prove that $\mathsf{wExtCom}$ satisfies the strong extractability with ε-simulation is related to an issue that is often referred to as *over-extraction* in the classical literature (e.g., [GLOV12, GGJS12, Kiy14]). Over-extraction means that an extractor may extract some non-$\bot$ message from an invalid commitment, instead of detecting the invalidness of the commitment. In particular, there does not exist a unique "committed message" when the commitment is ill-formed in $\mathsf{wExtCom}$, and extraction of such a non-unique message may collapse the committer's state. To deal with this issue, we have to add some mechanism which enables a receiver (and thus the extractor) to detect invalidness of the commitment.

One possible approach is to revisit the techniques developed in the classical setting, performing necessary surgery to make the proof work against QPT adversaries. However, as demonstrated by the above cited works, existing techniques in the classical setting are already delicate; even if it would work eventually, such a non-black-box treatment would further complicate the proof undesirably. Therefore, we present an alternative approach that deviates from existing ones in the classical setting; as we will show later, this new approach turns to be quantum-friendly.

Roughly speaking, our construction ExtCom works as follows:

Commit Stage:

1. The committer C generates shares $\{\mathsf{v}_i\}_{i=1}^n$ of a *verifiable secret sharing* (VSS) scheme (see Def. 1) of the message to be committed to, and then commits to each v_i using $\mathsf{wExtCom}$ separately in parallel.

[14] The corresponding message is well-defined (except for negligible probability) since we assume that Com is statistically binding.

2. C and the receiver R jointly run a "one-side simulatable" coin-flipping protocol based on $\mathsf{wExtCom}$ to generate a random subset T of $[n]$ of a certain size.[15] Specifically, they do the following:
   (a) R commits to a random string r_1 by $\mathsf{wExtCom}$.
   (b) C sends a random string r_2 in the clear.
   (c) R opens r_1. Then, both parties derive the subset T from $r_1 \oplus r_2$.
3. C opens the commitments corresponding to the subset T, and R checks their validness and consistency.

Decommit Stage:

1. C opens all the commitments. R checks those openings are valid. If they are valid, R runs the reconstruction algorithm of VSS to recover the committed message.

Using a similar argument as that for the soundness of the *MPC-in-the-head paradigm* [IKOS08, Goy11],[16] we can show that if a malicious committer passes the verification in the commit stage, then:

1. Most of the commitments of $\mathsf{wExtCom}$ generated in Step 1 are valid *as a commitment*; **and**
2. The committed shares in those valid commitments determines a *unique* message that can be recovered by the reconstruction algorithm of VSS.

Then, we can apply the special parallel weak extractability with ε-simulation of $\mathsf{wExtCom}$ to show the strong extractability with ε-simulation of ExtCom. We remark that essentially the same proof can also be used to show that the *parallel* execution of ExtCom is still strongly extractable with ε-simulation. We refer to this as the parallel-strong extractability with ε-simulation; it will play a critical role in our construction of ExtCom-and-Prove (see Sec. 2.2).

Dealing with Rejection in Commit Stage. So far, we have only focused on the case where the receiver accepts in the commit stage. However, the definition of (both weak and strong) extractability requires that the final state should be simulated even in the case where the receiver rejects in the commit stage. In this case, of course, the extractor does not need to extract anything, and thus the simulation is straightforward. A non-trivial issue, however, is that the extractor does not know if the receiver rejects in advance. This issue can be solved by a technique introduced in [BS20]. The idea is to just guess if the receiver accepts, and runs the corresponding extractor assuming that the guess is correct. This gives an intermediate extractor that succeeds with probability almost $1/2$ and its output correctly simulates the desired distribution conditioned on that it does not abort. Such an extractor can be compiled into a full-fledged extractor that does not abort by Watrous' rewinding lemma [Wat09].

Proof Idea for the Generalized Extract-and-Simulate Lemma. Finally, we briefly explain the idea for the proof of our generalized extract-and-simulate lemma. The basic idea is similar to the original extract-and-simulate lemma in [CCY21]—Use Jordan's lemma to decompose the adversary's internal state into "good" and "bad" subspaces, and amplify the extraction probability

[15] We remark that it is a non-trivial task to construct constant-round two-party coin-flipping from OWFs in the quantum setting, achieving the (even ε-)simulation-based security *against both parties*. Indeed, that will be one application of the strongly extractable commitment with ε-simulation, which we are now constructing. However, this is not a circular reasoning. Here, we need simulation-based security only against a malicious receiver. For such a one-side simulatable coin-flipping, the weakly extractable commitment $\mathsf{wExtCom}$ (with ε-simulation) suffices.

[16] To avoid disturbing the current discussion, we will provide more details of this argument in Sec. 2.2, where it is used again to establish the security of our ExtCom-and-Prove.

in the good subspace while effectively ignoring the bad-subspace components. However, the crucial difference is that in [CCY21], they define those subspaces with respect to the success probability of $\mathcal{A}$ whereas we define them with respect to the success probability of $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$. That is, for a noticeable δ, we apply Jordan's lemma to define a subspaces $S_{<\delta}$ and $S_{\geq\delta}$ such that

1. When $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$'s input is in $S_{<\delta}$ (resp. $S_{\geq\delta}$), it succeeds in extracting s^* with probability $< \delta$ (resp. $\geq \delta$).
2. Given a state in $S_{\geq\delta}$, we can extract s^* with overwhelming probability within $O(\delta^{-1})$ steps.
3. The above procedure does not cause any interference between $S_{<\delta}$ and $S_{\geq\delta}$.

We define $\mathcal{SE}$ to be an algorithm that runs the procedure in Item 2 and outputs s (which is supposed to be s^* in the case of success) and the post-execution state of $\mathcal{A}$. First, we consider simpler cases where the initial state of the experiments is a pure state $|\psi\rangle$ that is in either $S_{\geq\delta}$ or $S_{<\delta}$.

Case of $|\psi\rangle \in S_{\geq\delta}$: In this case, Item 2 implies that $\mathcal{SE}$ outputs s^* with overwhelming probability. In general, such an almost-deterministic quantum procedure can be done (almost) without affecting the state (e.g., see the *Almost-as-Good-as-New Lemma* in [Aar05, Lemma 2.2]). Therefore, $\mathsf{Exp}_{\mathsf{real}}$ and $\mathsf{Exp}_{\mathsf{ext}}$ are negligibly indistinguishable in this case.

Case of $|\psi\rangle \in S_{<\delta}$: For any state $|\psi_{<\delta}\rangle \in S_{<\delta}$, Item 1 implies

$$\Pr\left[\mathsf{Ext}_{\mathsf{Sim\text{-}less}}^{\mathcal{A}(|\psi_{<\delta}\rangle)} = s^*\right] \leq \delta.$$

On the other hand, our assumption (i.e., Inequality (1)) implies

$$\Pr\left[\mathsf{Ext}_{\mathsf{Sim\text{-}less}}^{\mathcal{A}(|\psi_{<\delta}\rangle)} = s^*\right] \geq (\Pr[\mathcal{A}(|\psi_{<\delta}\rangle) = \mathsf{Succ}])^c - \mathsf{negl}(\lambda)$$

for some constant c. By combining them, we have

$$\Pr[\mathcal{A}(|\psi_{<\delta}\rangle) = \mathsf{Succ}] \leq (\delta + \mathsf{negl}(\lambda))^{1/c}.$$

We note that the second output of $\mathcal{SE}$ in $\mathsf{Exp}_{\mathsf{ext}}$ is in $S_{<\delta}$ if the initial state is in $S_{<\delta}$ by Item 3. Therefore, if we run $\mathsf{Exp}_{\mathsf{real}}$ or $\mathsf{Exp}_{\mathsf{ext}}$ with an initial state in $S_{<\delta}$, it outputs Fail with probability $> 1 - (\delta + \mathsf{negl}(\lambda))^{1/c}$. Recall that when an experiment outputs Fail, no information about the internal state of $\mathcal{A}$ is revealed. Thus, the distinguishing advantage between those experiments can be bounded by $O(\delta^{1/c})$.

In general, the initial state is a superposition of $S_{<\delta}$ component and $S_{\geq\delta}$ component. Thanks to Item 3, we can reduce the general case to the above two cases. When doing that, there occurs an additional loss of the 4-th power of δ due to a technical reason (that appears in Lem. 8). Still, we can bound the distinguishing advantage between the two experiments by $O(\delta^{1/(4c)})$. This can be made to be an arbitrarily small noticeable function because δ is an arbitrarily small noticeable function. This suffices for establishing the ε-closeness of those experiments.

2.2 Black-Box ε-Simulatable ExtCom-and-Prove

Black-box zero-knowledge commit-and-prove allows a committer to commit to some message m (the Commit Stage), and later prove in zero-knowledge that the committed m satisfies some predicate

ϕ (the Prove Stage). What makes this primitive non-trivial is the requirement of black-box use of cryptographic building blocks; otherwise, this task can be fulfilled easily by giving a standard commitment to m first, and then running any zero-knowledge system over the commitment in a non-black-box manner.

MPC-in-the-Head. In the classical setting, black-box zero-knowledge commit-and-prove has been constructed following the so-called "MPC-in-the-head" paradigm [IKOS07, GLOV12]. To commit to m, the committer will imagine $n(\lambda)$ virtual parties "in his head", who jointly execute a (n, t)-*verifiable secret sharing* (VSS) scheme to share the message m. Roughly speaking, such a VSS scheme ensures that if only $\leq t$ parties are corrupted, then all the honest parties will learn their shares properly and can always recover the m by exchanging their shares if they want; however, the $\leq t$ number of corrupted parties learns no information about m. Denote the views of the n virtual parties during the VSS sharing stage execution as $\{\mathsf{v}_i\}_{i\in[n]}$. To finish the Commit Stage, the committer commits to these n views in parallel, using independent instances of a statistically-binding commitment Com for each view separately.

To prove the committed m satisfies a predicate ϕ, the committer continues by asking the n virtual parties to execute an (n, t)-secure MPC with $\{\mathsf{v}_i\}_{i\in[n]}$ as their respective input; this MPC is executed for a functionality that collects all the views from each party, runs the VSS reconstruction algorithm to recover the value m, and finally outputs $\phi(m)$ to each party. Denote the views of the n parties during this MPC execution as $\{\mathsf{v}'_i\}_{i\in[n]}$. The committer commits to $\{\mathsf{v}'_i\}_{i\in[n]}$ in parallel, using independent instances of a statistically-binding commitment for each v'_i separately. Next, the committer and receiver will execute a coin-flipping protocol to determine a random size-t subset $T \subseteq [n]$; the committer decommits to $\{\mathsf{v}_i\}_{i\in T}$ and $\{\mathsf{v}'_i\}_{i\in T}$, and the receiver accepts if and only if these views are *consistent* w.r.t. the VSS and MPC execution. Roughly speaking, this means that for each pair $i, j \in T$, v_i and v_j (resp. v'_i and v'_j) contain consistent incoming/outgoing messages, and they are the messages computed following the honest parties' algorithm w.r.t. the VSS (resp. the MPC) protocol.

To see why this construction is zero-knowledge, observe that a simulator $\mathcal{S}$ can pre-decide a size-t set $\widetilde{T} \subseteq [n]$ and commit to "fake" consistent views for both the VSS and MPC execution for parties in set $\widetilde{T}$, *without knowing the actual value* m. This can be done because both the VSS and MPC reveals no information if only t views are leaked. For the views in $[n] \setminus \widetilde{T}$, $\mathcal{S}$ simply commits to all-0 strings of proper length. Then, $\mathcal{S}$ will bias the coin-flipping result to $\widetilde{T}$ using the simulator for the coin-flipping protocol against the malicious receiver. In this way, only the faked views in the set $\widetilde{T}$ need to be revealed, and they will pass the receiver's consistency check as they are faked to be consistent.

Soundness can be proven as follows. Due to the security property of the VSS and MPC, corrupting $\leq t$ parties during the execution will not change the value $\phi(m)$ learned by the $\geq n - t$ honest parties. Therefore, to lie about $\phi(m)$, there must be $> t$ number of virtual parties being corrupted. However, this will necessarily yield many inconsistent pairs of views. Since the coin-flipping step determines a size-t (pseudo-)random subset $T \subseteq [n]$ for consistency check, the inconsistent views will be caught by the receiver, except with negligible probability (by setting n and t properly). We remark that the actual proof for soundness requires a more involved argument to formalize the above intuition (See Lem. 20 for details).

From the above discussion, it should be clear that the coin-flipping must achieve simulation-based security against corrupted receivers (for zero-knowledge), while it only needs to be indistinguishability-based (IND) secure against corrupted committer (for soundness).

Our Construction. Our construction follows the above paradigm with the following modifications. To make the commitment stage extractable, we ask the committer to use the ε-simulatable

(strongly) extractable commitment ExtCom we constructed in Sec. 2.1, in place of Com. As mentioned in Sec. 2.1, ExtCom maintains its ε-simulatable extractability even when used in parallel; therefore, if suffices for the committer's parallel commitments to the views. For a malicious QPT committer, the parallel-strong extractability of ExtCom allows us to extract the committed value while performing a ε-simulation for the committer's post-extraction state. Thus, we obtain the ε-simulatable extractability of the Commit Stage.

To obtain soundness and ε-ZK against QPT adversaries, we only need to build a coin-flipping that is *ε-simulatable against QPT malicious receiver, and IND-secure against QPT malicious committer*, to replace the classical coin-flipping protocol used to determine T. Once we have such a coin-flipping, soundness and ε-ZK can be established as in the above classical setting.

We construct such a coin-flipping using the $\mathsf{wExtCom}$ constructed in Sec. 2.1 (note that we do not need the strongly extractable ExtCom). This protocol is the same as Step 2 of our $\mathsf{wExtCom}$-to-ExtCom compiler:

1. The receiver uses $\mathsf{wExtCom}$ to commit to a random string r_1;
2. The committer samples and sends a random string r_2;
3. The receiver decommits to r_1. The coin-flipping result is determined as $r := r_1 \oplus r_2$, which determines the size-t subset T.

The pseudo-randomness of r against the corrupted committer follows from the computationally-hiding property of $\mathsf{wExtCom}$. This allows us to prove soundness. To simulate for a corrupted receiver (i.e., to bias the coin-fliping result to a random $\widetilde{r}$ that $\mathcal{S}$ obtained from the ideal-world functionality), $\mathcal{S}$ simply runs the ε-simulation extractor to extract the string r_1^* committed in $\mathsf{wExtCom}$ by the malicious receiver, and set $r_2 := \widetilde{r} \oplus r_1^*$. Note that the extractor is also a ε-simulator for the post-extraction state; thus, we obtain ε-zero-knowledge.

2.3 Black-Box ε-Simulatable 2PC

At a high level, our construction consists of the following 3 steps:

1. There exists a known constant-round black-box compiler *in the post-quantum setting*, which converts bounded-parallel string-OTs between two parties to general-purpose 2PC [IPS08].
2. There also exists a constant-round black-box compiler in the post-quantum setting, which converts semi-honest bit-OTs to bounded-parallel string-OTs between two parties, in the $\mathcal{F}_{\text{SO-COM}}^{t}$-hybrid model [CDMW09, GLSV21]. $\mathcal{F}_{\text{SO-COM}}^{t}$ is a two-party ideal functionality formalized recently in [GLSV21]: it allows a committer to commit to an a-priori fixed polynomial number $t(\lambda)$ of messages in parallel, and later decommit to a subset of these commitments named by the receiver (thus, "SO" stands for "selectively opening").
3. Therefore, all we need to do is to construct a constant-round black-box ε-simulatable protocol that implements $\mathcal{F}_{\text{SO-COM}}^{t}$, and then rely on the "non-concurrent composition" lemma in the stand-alone setting (developed in [Can00]) to obtain the bounded-parallel two-party string-OTs (and eventually 2PC via Step 1). In the following, we show how to obtain such a realization of $\mathcal{F}_{\text{SO-COM}}^{t}$ using the ExtCom-and-Prove from Sec. 2.2.

Remark 1. Steps 1 and 2 have been used to construct black-box and constant-round *stand-alone secure* 2PC (actually, even MPC) protocols in the classical setting [PW09, CDMW09, Goy11]. However, to our best knowledge, these two steps have not been explicitly and rigorously formalized *in the stand-alone setting*. There are some subtleties regarding the composition of protocols that

require extra caution. One reason for the lack of a rigorous treatment is that the main focus of [IPS08, CDMW09] is the universally-composable (UC) model [Can01]. The recent work [GLSV21] made it explicit by formally defining and constructing bounded-parallel two-party string-OTs and the $F^t_{\text{SO-COM}}$ functionality. We provide further clarification of these subtleties in Sec. 7.3.

To construct an ε-simulatable protocol for $\mathcal{F}^t_{\text{SO-COM}}$, we ask the committer to commit to his t messages $(m_1, \ldots, m_t)$ using the Commit Stage of the ExtCom-and-Prove. To decommit to the subset of messages determined by the receiver's challenge set $I \subseteq [n]$, the committer first sends $\{m_i\}_{i\in I}$ to the receiver; then, both parties execute the Prove Stage, where the committer proves the following predicate:

$$\phi_{I,\{m_i\}_{i\in I}}(x) = \begin{cases} 1 & \text{if } (x = m'_1 \| \ldots \| m'_t) \wedge (\forall i \in I, m'_i = m_i) \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The proof of security is straightforward: if the committer is corrupted, simulation can be done via the ε-simulatable extractability of the Commit Stage; if the receiver is corrupted, simulation can be done via the ε-ZK property of the ExtCom-and-Prove.

Here is one caveat: in Step 3, we actually need to rely on a variant of the non-concurrent composition lemma by Canetti [Can00], which should:

1. hold in the post-quantum setting. That is, it should hold for classical protocols secure against QPT adversaries; **and**
2. allow composition of *ε-simulatable* protocols. Roughly speaking, it requires that if $\pi^{\mathcal{F}}$ is a post-quantum ε-simulatable protocol for an ideal functionality $\mathcal{G}$ in the $\mathcal{F}$-hybrid model and ϕ is a post-quantum ε-simulatable protocol for the ideal functionality $\mathcal{F}$ in the plain model, then replacing the $\mathcal{F}$ calls in $\pi^{\mathcal{F}}$ with ϕ yields a post-quantum ε-simulatable protocol for $\mathcal{G}$ in the plain model, as long as the execution of ϕ is not interleaved with other parts of $\pi^{(\cdot)}$.

Fortunately, these requirements can be obtained by generalizing the proof of the non-concurrent composition lemma in [Can00] to the ε-simulatable security against QPT adversaries. We present a formal treatment of it in Sec. 7.2.

3 Preliminaries

Basic Notations. We denote by λ the security parameter throughout the paper. For a positive integer $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, ..., n\}$. For a finite set $\mathcal{X}$, $x \leftarrow \mathcal{X}$ means that x is uniformly chosen from $\mathcal{X}$.

A function $f : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all polynomial p and sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$; it is said to be *overwhelming* if $1 - f$ is negligible, and said to be *noticeable* if there is a polynomial p such that $f(\lambda) \geq 1/p(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. We denote by poly an unspecified polynomial and by negl an unspecified negligible function.

We use PPT and QPT to mean (classical) probabilistic polynomial time and quantum polynomial time, respectively. For a classical probabilistic or quantum algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ means that $\mathcal{A}$ is run on input x and outputs y. An adversary (or malicious party) is modeled as a non-uniform QPT algorithm $\mathcal{A}$ (with quantum advice) that is specified by a sequence of polynomial-size quantum circuits and quantum advices $\{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$. In an execution with the security parameter λ, $\mathcal{A}$ runs $\mathcal{A}_\lambda$ taking ρ_λ as the advice. We often omit the index λ and just write $\mathcal{A}(\rho)$ to mean a non-uniform QPT algorithm specified by $\{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ for simplicity.

We use the bold font (like $\mathbf{X}$) to denote quantum registers. For a quantum state ρ, $\|\rho\|_{tr}$ denotes the trace norm of ρ.

3.1 Quantum Computation

Interactive Quantum Machines and Oracle-Aided Quantum Machines. We rely on the definition of interactive quantum machines and oracle-aided quantum machines that are given oracle access to an interactive quantum machine, following [Unr12]. Roughly, an interactive quantum machine $\mathcal{A}$ is formalized by a unitary over registers $\mathbf{M}$ for receiving and sending messages, and $\mathbf{A}$ for maintaining $\mathcal{A}$'s internal state. For two interactive quantum machines $\mathcal{A}$ and $\mathcal{B}$ that share the same message register $\mathbf{M}$, an interaction between $\mathcal{A}$ and $\mathcal{B}$ proceeds by alternating invocations of $\mathcal{A}$ and $\mathcal{B}$ while exchanging messages over $\mathbf{M}$.

An oracle-aided quantum machine $\mathcal{S}$ given oracle access to an interactive quantum machine $\mathcal{A}$ with an initial internal state ρ (denoted by $\mathcal{S}^{\mathcal{A}(\rho)}$) is allowed to apply the unitary part of $\mathcal{A}$ (the unitary obtained by deferring all measurements by $\mathcal{A}$ and omitting these measurements) and its inverse in a black-box manner. $\mathcal{S}$ is only allowed to act on $\mathcal{A}$'s internal register $\mathbf{A}$ through oracle access. We refer to [Unr12] for formal definitions of interactive quantum machines and black-box access to them.

Indistinguishability of Quantum States. We define computational and statistical indistinguishability of quantum states similarly to [BS20, CCY21].

We may consider random variables over bit strings or over quantum states. This will be clear from the context. For ensembles of random variables $\mathcal{X} = \{X_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ and $\mathcal{Y} = \{Y_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ over the same set of indices $I = \bigcup_{\lambda\in\mathbb{N}} I_\lambda$ and a function δ, we use $\mathcal{X} \overset{\text{c}}{\approx}_\delta \mathcal{Y}$ to mean that for any non-uniform QPT algorithm $\{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}_\lambda(X_i, \rho_\lambda)] - \Pr[\mathcal{A}_\lambda(Y_i, \rho_\lambda)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).$$

In particular, when the above holds for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, and simply write $\mathcal{X} \overset{\text{c}}{\approx} \mathcal{Y}$. Unless stated differently, throughout this paper, computational indistinguishability is always w.r.t. non-uniform QPT adversaries.

Similarly, we use $\mathcal{X} \overset{\text{s}}{\approx}_\delta \mathcal{Y}$ to mean that for any unbounded time algorithm $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}(X_i)] - \Pr[\mathcal{A}(Y_i)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).^{17}$$

In particular, when the above hold for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are statistically indistinguishable, and simply write $\mathcal{X} \overset{\text{s}}{\approx} \mathcal{Y}$. Moreover, we write $\mathcal{X} \equiv \mathcal{Y}$ to mean that X_i and Y_i are distributed identically for all $i \in I$

When we consider an ensemble $\mathcal{X}$ that is only indexed by λ, (i.e., $I_\lambda = \{\lambda\}$), we write $\mathcal{X} = \{X_\lambda\}_\lambda$ for simplicity.

3.2 Technical Lemmas

Serfling's Inequality. The following *two-sided* version of Serfling's inequality is taken from [BF10].

[17] In other words, $\mathcal{X} \overset{\text{s}}{\approx}_\delta \mathcal{Y}$ means that there exists a negligible function $\mathsf{negl}(\cdot)$ such that the trace distance between ρ_{X_i} and ρ_{Y_i} is at most $\delta(\lambda) + \mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$ and $i \in I_\lambda$ where ρ_{X_i} and ρ_{Y_i} denote the density matrices corresponding to X_i and Y_i.

Lemma 1 (Serfling's Inequality [Ser74, BF10]). *Let $\mathbf{b} \in \{0,1\}^n$ be a bit string with $\mu \cdot n$ non-zero bits (i.e., a μ-fraction is 1's). Let the random variables $(Y_1, Y_2, \ldots, Y_k)$ be obtained by sampling k random entries from $\mathbf{b}$* without replacement. *Let $\bar{Y} := \frac{1}{k}\sum_i^k Y_i$. Then, for any $\delta > 0$, it holds that*

$$\Pr\left[|\bar{Y} - \mu| > \delta\right] \leq 2\exp\left(\frac{-2\delta^2 kn}{n-k+1}\right).$$

Watrous' Rewinding Lemma. The following is Watrous' rewinding lemma [Wat09] in the form of [BS20, Lemma 2.1].

Lemma 2 (Watrous' Rewinding Lemma [Wat09]). *There is a quantum algorithm R that gets as input the following:*

- *A quantum circuit Q that takes n-input qubits in register Inp and outputs a classical bit b (in a register outside Inp) and an m-qubit output.*
- *An n-qubit state ρ in register Inp.*
- *A number $T \in \mathbb{N}$ in unary.*

$\mathsf{R}(1^T, \mathsf{Q}, \rho)$ executes in time $T \cdot |\mathsf{Q}|$ and outputs a distribution over m-qubit states $D_\rho := \mathsf{R}(1^T, \mathsf{Q}, \rho)$ with the following guarantees.

For an n-qubit state ρ, denote by Q_ρ the conditional distribution of the output distribution $\mathsf{Q}(\rho)$, conditioned on $b = 0$, and denote by $p(\rho)$ the probability that $b = 0$. If there exist $p_0, q \in (0,1)$, $\gamma \in (0, \frac{1}{2})$ such that:

- *Amplification executes for enough time: $T \geq \frac{\log(1/\gamma)}{4p_0(1-p_0)}$,*
- *There is some minimal probability that $b = 0$: For every n-qubit state ρ, $p_0 \leq p(\rho)$,*
- *$p(\rho)$ is input-independent, up to γ distance: For every n-qubit state ρ, $|p(\rho) - q| < \gamma$, and*
- *q is closer to $\frac{1}{2}$: $p_0(1-p_0) \leq q(1-q)$,*

then for every n-qubit state ρ,

$$\mathsf{TD}(\mathsf{Q}_\rho, D_\rho) \leq 4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)}.$$

Unruh's Rewinding Lemma. The following lemma is proven in [Unr12].

Lemma 3 (Unruh's Rewinding Lemma [Unr12, Lemma 7]). *Let C be a finite set. Let $\{\Pi_i\}_{i\in C}$ be orthogonal projectors on a Hilbert space $\mathcal{H}$. Let $|\psi\rangle \in \mathcal{H}$ be a unit vector. Let $\alpha := \sum_{i\in C}\frac{1}{|C|}\|\Pi_i|\psi\rangle\|^2$ and $\beta := \sum_{i,j\in C}\frac{1}{|C|^2}\|\Pi_i\Pi_j|\psi\rangle\|^2$ Then we have $\beta \geq \alpha^3$.*

3.3 Verifiable Secret Sharing Schemes

We present in Def. 1 the definition of verifiable secret sharing (VSS) schemes [CGMA85]. We remark that [BGW88, CDD+99] implemented $(n+1, \lfloor n/3 \rfloor)$-perfectly secure VSS schemes. These constructions suffice for all the applications in the current paper.

Definition 1 (Verifiable Secret Sharing). *An $(n+1, t)$-perfectly secure VSS scheme Π_{VSS} consists of a pair of protocols $(\mathsf{VSS}_{\mathsf{Share}}, \mathsf{VSS}_{\mathsf{Recon}})$ that implement respectively the sharing and reconstruction phases as follows.*

– **Sharing Phase $\mathsf{VSS}_{\mathsf{Share}}$:** *Player P_{n+1} (referred to as dealer) runs on input a secret s and randomness r_{n+1}, while any other player P_i ($i \in [n]$) runs on input a randomness r_i. During this phase players can send (both private and broadcast) messages in multiple rounds.*

– **Reconstruction Phase $\mathsf{VSS}_{\mathsf{Recon}}$:** *Each shareholder sends its view v_i ($i \in [n]$) of the Sharing Phase to each other player, and on input the views of all players (that can include bad or empty views) each player outputs a reconstruction of the secret s.*

All computations performed by honest players are efficient. The computationally unbounded adversary can corrupt up to t players that can deviate from the above procedures. The following security properties hold.

1. **Perfectly Verifiable-Committing:** *if the dealer is dishonest, then one of the following two cases happen (i.e., with probability 1):*
   (a) *During the Sharing Phase, honest players disqualify the dealer, therefore they output a special value $\perp$ and will refuse to play the reconstruction phase;*
   (b) *During the Sharing Phase, honest players do not disqualify the dealer. Therefore such a phase determines a unique value s^* that belongs to the set of possible legal values that does not include $\perp$, which will be reconstructed by the honest players during the reconstruction phase.*
2. **Secrecy:** *if the dealer is honest, then the adversary obtains no information about the shared secret before running the protocol* Recon*. More accurately, there exists a PPT oracle machine $\mathcal{S}^{(\cdot)}$ such that for any message m, and every (potentially inefficient) adversary $\mathcal{A}$ corrupting a set T of parties with $|T| \leq t$ during the Sharing Phase $\mathsf{VSS}_{\mathsf{Share}}(m)$ (denote $\mathcal{A}$'s view in this execution as $\mathsf{View}_{\mathcal{A},T}(1^\lambda, m)$), the following holds: $\{\mathsf{View}_{\mathcal{A},T}(1^\lambda, m)\} \stackrel{i.d.}{\equiv} \{\mathcal{S}^{\mathcal{A}}(1^\lambda, T)\}$.*
3. **Correctness:** *if the dealer is honest throughout the protocols, then each honest player will output the shared secret s at the end of protocol* Recon*.*

3.4 Information-Theoretic MPC and the MPC-in-the-Head Paradigm

We first recall *information-theoretically secure* MPC and relevant notions that will be employed in the MPC-in-the-head paradigm shown later.

Information-Theoretic MPC. We now define MPC in the information-theoretic setting (i.e., secure against unbounded adversaries).

Definition 2 (Perfectly/Statistically-Secure MPC). *Let $f : (\{0,1\}^*)^n \mapsto (\{0,1\}^*)^n$ be an n-ary functionality, and let Π be a protocol. We say that Π (n,t)-perfectly* (resp., statistically) securely *computes f if for every static, malicious, and (possibly-inefficient) probabilistic adversary $\mathcal{A}$ in the real model, there exists a probabilistic adversary $\mathcal{S}$ of comparable complexity (i.e., with running time polynomial in that of $\mathcal{A}$) in the ideal model, such that for every $I \subset [n]$ of cardinality at most t, every $\boldsymbol{x} = (x_1, \ldots, x_n) \in (\{0,1\}^*)^n$ (where $|x_1| = \cdots = |x_n|$), and every $z \in \{0,1\}^*$, it holds that:*

$$\{\mathsf{REAL}_{\Pi,\mathcal{A}(z),I}(\boldsymbol{x})\} \stackrel{i.d.}{\equiv} \{\mathsf{IDEAL}_{f,\mathcal{S}(z),I}(\boldsymbol{x})\} \quad \big(\textit{resp.}, \{\mathsf{REAL}_{\Pi,\mathcal{A}(z),I}(\boldsymbol{x})\} \stackrel{s}{\approx} \{\mathsf{IDEAL}_{f,\mathcal{S}(z),I}(\boldsymbol{x})\}\big).$$

Recall that the MPC protocol from [BGW88] achieves (n,t)-perfect security (against static and malicious adversaries) with t being a constant fraction of n.

Theorem 1 ([BGW88]). *Consider a synchronous network with pairwise private channels. Then, for every n-ary functionality f, there exists a protocol that (n,t)-perfectly securely computes f in the presence of a static malicious adversary for any $t < n/3$.*

Consistency, Privacy, and Robustness. We now define some notation related to MPC protocols. Their roles will become clear when we discuss the MPC-in-the-head technique later.

Definition 3 (View Consistency). *A view* View_i *of an honest player* P_i *during an MPC computation* Π *contains input and randomness used in the computation, and all messages received from and sent to the communication tapes. A pair of views* $(\mathsf{View}_i, \mathsf{View}_j)$ *is* consistent *with each other if*

1. *Both corresponding players* P_i *and* P_j *individually computed each outgoing message honestly by using the random tapes, inputs and incoming messages specified in* View_i *and* View_j *respectively, and:*
2. *All output messages of* P_i *to* P_j *appearing in* View_i *are consistent with incoming messages of* P_j *received from* P_i *appearing in* View_j *(and vice versa).*

Remark 2 (View Consistency of VSS). Although Def. 3 defines view consistency for MPC protocols, we will also refer to the view consistency for the execution of verifiable secret sharing schemes (Def. 1). The views $(\mathsf{v}_i, \mathsf{v}_j)$ of players i and j (excluding the dealer) during the execution of $\mathsf{VSS}_{\mathsf{Share}}$ is said to be consistent if any only if $(\mathsf{v}_i, \mathsf{v}_j)$ satisfies the two requirements in Def. 3.

We further define the notions of correctness, privacy, and robustness for multi-party protocols.

Definition 4 (Semi-Honest Computational Privacy). *Let* $1 \le t < n$, *let* Π *be an MPC protocol, and let* $\mathcal{A}$ *be any* static, PPT, and semi-honest *adversary. We say that* Π *realizes a function* $f : (\{0,1\}^*)^n \mapsto (\{0,1\}^*)^n$ *with* semi-honest (n,t)-computational privacy *if there is a PPT simulator* $\mathcal{S}$ *such that for any inputs* $x, w_1, \ldots, w_n$, *every subset* $T \subset [n]$ $(|T| \le t)$ *of players corrupted by* $\mathcal{A}$, *and every* D *with circuit size at most* $\mathsf{poly}(\lambda)$, *it holds that*

$$\left| \Pr[D(\mathsf{View}_T(x, w_1, \ldots, w_n)) = 1] - \Pr[D(\mathcal{S}(T, x, \{w_i\}_{i \in T}, f_T(x, w_1, \ldots, w_n))) = 1] \right| \le \mathsf{negl}(\lambda), \quad (3)$$

where $\mathsf{View}_T(x, w_1, \ldots, w_n)$ *is the joint view of all players.*

Definition 5 (Statistical/Perfect Correctness). *Let* Π *be an MPC protocol. We say that* Π *realizes a deterministic n-party functionality* $f(x, w_1, \ldots, w_n)$ *with* perfect (resp., statistical) *correctness if for all inputs* $x, w_1, \ldots, w_n$, *the probability that the output of some party is different from the output of some party is different from the actual output of* f *is 0 (resp., negligible in k), where the probability is over the independent choices of the random inputs* $r_1, \ldots, r_n$ *of these parties.*

Definition 6 (Perfect/Statistical Robustness). *Assume the same setting as the previous definition. We say that* Π *realizes* f *with* (n,t)-perfect (resp., statistical) robustness *if in addition to being perfectly (resp., statistical) correct in the presence of a semi-honest adversary as above, it enjoys the following* robustness *property against any computationally unbounded malicious adversary corrupting a set* T *of at most* t *parties, and for any inputs* $(x, w_1, \ldots, w_n)$*: if there is no* $(w'_1, \ldots, w'_n)$ *such that* $f(x, w'_1, \ldots, w'_n) = 1$, *then the probability that some uncorrupted player outputs 1 in an execution of* Π *in which the inputs of the honest parties are consistent with* $(x, w_1, \ldots, w_n)$ *is 0 (resp., negligible in λ).*

MPC-in-the-Head. MPC-in-the-head is a technique developed for constructing black-box ZK protocols from MPC protocols [IKOS07]. Intuitively, the MPC-in-the-head idea works as follows. Let $\mathcal{F}_{\text{ZK}}$ be the zero-knowledge functionality for an **NP** language. Assume there are n parties holding

a witness in a secret-sharing form. $\mathcal{F}_{\text{ZK}}$ takes as public input x and one share from each party, and outputs 1 iff the secret reconstructed from the shares is a valid witness. To build a ZK protocol, the prover runs in his head an execution of MPC w.r.t. $\mathcal{F}_{\text{ZK}}$ among n imaginary parties, each one participating in the protocol with a share of the witness. Then, it commits to the view of each party separately. The verifier obtains t randomly chosen views, checks that such views are "consistent" (see Def. 3), and accepts if the output of every party is 1. The idea is that, by selecting the t views at random, V will catch inconsistent views if the prover cheats.

We emphasize that, in this paradigm, a malicious prover decides the randomness of each virtual party, including those not checked by the verifier (corresponding to honest parties in the MPC execution). Therefore, MPC protocols with standard computational security may fail to protect against such attacks. We need to ensure that the adversary cannot force a wrong output even if it additionally controls the honest parties' random tapes. The $(n, \lfloor n/3 \rfloor)$-perfectly secure MPC protocol in Thm. 1 suffices for this purpose (see also Rmk. 3).

One can extend this technique further (as in [GLOV12]), to prove a general predicate ϕ about an arbitrary value α. Namely, one can consider the functionality $\mathcal{F}_\phi$ in which party i participates with input a VSS share $[\alpha]_i$. $\mathcal{F}_\phi$ collects all such shares, and outputs 1 iff $\phi(\mathsf{VSS}_{\mathsf{Recon}}([\alpha]_1, \ldots, [\alpha]_n)) = 1$.

Remark 3 (Exact Security Requirements on the Underlying MPC.). To be more accurate, any MPC protocol that achieves *semi-honest* (n,t)*-computational privacy (as per Def. 4) and* (n,t)*-perfect robustness (as per Def. 6)* will suffice for the MPC-in-the-head application.[18] These two requirements are satisfied by any (n,t)-perfectly secure MPC (and, in particular, the one from Thm. 1).

3.5 Post-Quantum Extractable Commitment

We give a definition of post-quantum (strongly) extractable commitments with ε-simulation. We will omit the security parameter from the input to parties when it is clear from the context.

Definition 7 (Post-Quantum Commitment). *A* post-quantum commitment scheme Π *is a classical interactive protocol between interactive* PPT *machines C and R. Let $m \in \{0,1\}^{\ell(\lambda)}$ (where $\ell(\cdot)$ is some polynomial) is a message that C wants to commit to. The protocol consists of the following stages:*

- **Commit Stage:** *$C(m)$ and R interact with each other to generate a transcript (which is also called a commitment) denoted by com,[19] C's state ST_C, and R's output $b_{\text{com}} \in \{0,1\}$ indicating acceptance (i.e., $b_{\text{com}} = 1$) or rejection (i.e., $b_{\text{com}} = 0$). We denote this execution by $(\mathsf{com}, \mathsf{ST}_C, b_{\text{com}}) \leftarrow \langle C(m), R\rangle(1^\lambda)$. When C is honest, ST_C is classical, but when we consider a malicious quantum committer $C^*(\rho)$, we allow it to generate any quantum state ST_{C^*}. Similarly, a malicious quantum receiver $R^*(\rho)$ can output any quantum state, which we denote by OUT_{R^*} instead of b_{com}.*
- **Decommit Stage:** *C generates a decommitment decom from ST_C. We denote this procedure by $\mathsf{decom} \leftarrow C(\mathsf{ST}_C)$.[20] Then it sends a message m and decommitment decom to R, and R outputs a bit $b_{\text{dec}} \in \{0,1\}$ indicating acceptance (i.e., $b_{\text{dec}} = 1$) or rejection (i.e., $b_{\text{dec}} = 0$). We assume that R's verification procedure is deterministic and denote it by $\mathsf{Verify}(\mathsf{com}, m, \mathsf{decom})$.[21] W.l.o.g., we assume that R always rejects (i.e., $\mathsf{Verify}(\mathsf{com}, \cdot, \cdot) = 0$) whenever $b_{\text{com}} = 0$. (Note that w.l.o.g., com can include b_{com} because we can always modify the protocol to ask R to send b_{com} as the last round message.)*

[18] It is also worth noting that the (n,t)-perfect robustness could be replaced with *adaptive* (n,t)*-statistical robustness*. See [IKOS07, Section 4.2] for more details.

[19] That is, we regard the whole transcript as a commitment.

[20] We could define ST_C to be decom itself w.l.o.g. However, we define them separately because this is more convenient when we define ExtCom-and-Prove in Def. 17, which is an extension of post-quantum extractable commitments.

[21] Note that Verify is well-defined since our syntax does not allow R to keep a state from the commit stage.

The scheme satisfies the following correctness requirement:

1. **Correctness.** *For any* $m \in \{0,1\}^{\ell(\lambda)}$*, it holds that*

$$\Pr\left[b_{\mathrm{com}} = b_{\mathrm{dec}} = 1 : \begin{array}{c}(\mathsf{com}, \mathsf{ST}_C, b_{\mathrm{com}}) \leftarrow \langle C(m), R\rangle(1^\lambda) \\ \mathsf{decom} \leftarrow C(\mathsf{ST}_C) \\ b_{\mathrm{dec}} \leftarrow \mathsf{Verify}(\mathsf{com}, m, \mathsf{decom})\end{array}\right] = 1.$$

Definition 8 (Computationally Hiding). *A post-quantum commitment* Π *is* computationally hiding *if for any* $m_0, m_1 \in \{0,1\}^{\ell(\lambda)}$ *and any non-uniform QPT receiver* $R^*(\rho)$*, the following holds:*

$$\{\mathsf{OUT}_{R^*} : (\mathsf{com}, \mathsf{ST}_C, \mathsf{OUT}_{R^*}) \leftarrow \langle C(m_0), R^*(\rho)\rangle(1^\lambda)\}_\lambda \stackrel{c}{\approx} \{\mathsf{OUT}_{R^*} : (\mathsf{com}, \mathsf{ST}_C, \mathsf{OUT}_{R^*})\langle C(m_1), R^*(\rho)\rangle(1^\lambda)\}_\lambda.$$

Definition 9 (Statistically Binding). *A post-quantum commitment* Π *is* statistically binding *if for any unbounded-time comitter* C^**, the following holds:*

$$\Pr\left[\begin{array}{l}\exists\ m, m', \mathsf{decom}, \mathsf{decom}',\ m \neq m' \\ \wedge\ \mathsf{Verify}(\mathsf{com}, m, \mathsf{decom}) = \mathsf{Verify}(\mathsf{com}, m', \mathsf{decom}') = 1\end{array} : (\mathsf{com}, \mathsf{ST}_{C^*}, b_{\mathrm{com}}) \leftarrow \langle C^*, R\rangle(1^\lambda)\right] = \mathsf{negl}(\lambda).$$

Definition 10 (Committed Values). *For a post-quantum commitment* Π*, we define the value function as follows:*

$$\mathsf{val}_\Pi(\mathsf{com}) := \begin{cases} m & \text{if } \exists \text{ unique } m \text{ s.t. } \exists\ \mathsf{decom}, \mathsf{Verify}(\mathsf{com}, m, \mathsf{decom}) = 1 \\ \bot & \text{otherwise}\end{cases}.$$

We say that com *is valid if* $\mathsf{val}_\Pi(\mathsf{com}) \neq \bot$ *and invalid if* $\mathsf{val}_\Pi(\mathsf{com}) = \bot$*.*

Then we give the definition of the strong extractability with ε-simulation. The definition is similar to that of post-quantum extractable commitments in [BS20, BLS21] except that we allow an (arbitrarily small) noticeable approximation error similarly to post-quantum ε-zero-knowledge [CCY21]. We note that we call it the *strong* extractability since we also define a weaker version of extractability in Def. 13 in Sec. 5.1.

Definition 11 (Strong Extractability with ε-Simulation). *A commitment scheme* Π *is* strongly extractable with ε-simulation *if there exists a QPT algorithm* $\mathcal{SE}$ *(called the* ε*-simulation strong-extractor) such that for any noticeable* $\varepsilon(\lambda)$ *and any non-uniform QPT* $C^*(\rho)$*,*

$$\left\{\mathcal{SE}^{C^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})\right\}_\lambda \stackrel{c}{\approx}_\varepsilon \left\{(\mathsf{val}_\Pi(\mathsf{com}), \mathsf{ST}_{C^*}) : (\mathsf{com}, \mathsf{ST}_{C^*}, b_{\mathrm{com}}) \leftarrow \langle C^*(\rho), R\rangle(1^\lambda)\right\}_\lambda.$$

We also define the parallel version.

Definition 12 (Parallel-Strong Extractability with ε-Simulation). *A commitment scheme* Π *is* parallelly strongly extractable with ε-simulation *if for any integer* $n = \mathsf{poly}(\lambda)$*, there exists a QPT algorithm* $\mathcal{SE}_{\mathsf{par}}$ *(called the* ε*-simulation parallel-strong-extractor) such that for any noticeable* $\varepsilon(\lambda)$ *and any non-uniform QPT* $C^*(\rho)$*,*

$$\begin{aligned}&\left\{\mathcal{SE}_{\mathsf{par}}^{C^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})\right\}_\lambda \\ \stackrel{c}{\approx}_\varepsilon &\left\{(\Lambda_{\{b_{\mathrm{com},j}\}_{j=1}^n}(\{\mathsf{val}(\mathsf{com}_j)\}_{j=1}^n), \mathsf{ST}_{C^*}) : (\{\mathsf{com}_j\}_{j=1}^n, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^n) \leftarrow \langle C^*(\rho), R^n\rangle(1^\lambda)\right\}_\lambda\end{aligned}$$

where $(\{\mathsf{com}_j\}_{j=1}^n, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^n) \leftarrow \langle C^*(\rho), R^n\rangle(1^\lambda)$ *means that* $C^*(\rho)$ *interacts with* n *copies of the honest receiver* R *in parallel and the execution results in transcripts* $\{\mathsf{com}_j\}_{j=1}^n$*, the final state* ST_{C^*}*, and outputs* $\{b_{\mathrm{com},j}\}_{j=1}^n$ *of each copy of* R *and*

$$\Lambda_{\{b_{\mathrm{com},j}\}_{j=1}^n}(\{\mathsf{val}(\mathsf{com}_j)\}_{j=1}^n) := \begin{cases} \{\mathsf{val}_\Pi(\mathsf{com}_j)\}_{j=1}^n & \text{if } \forall\ j \in [n]\ b_{\mathrm{com},j} = 1 \\ \bot & \text{otherwise} \end{cases}.$$

Remark 4. We remark that the above definition only requires the extractor to extract the committed values when R accepts in all the parallel sessions. In particular, when R accepts in some sessions but not in others, the extractor does not need to extract the committed values at all. An alternative stronger (and probably more natural) definition would require the extractor to extract $\mathsf{val}_\Pi(\mathsf{com}_j)$ for all $j \in [n]$ such that R accepts in the j-th session. But we define it in the above way since it suffices for our purpose and we do not know if our construction satisfies the stronger one.

4 Extract-and-Simulate Lemma

We prove a lemma that can be seen as an ε-simulation variant of Unruh's rewinding lemma (Lem. 3) in typical applications. This lemma is the technical core of all the results in this paper.

4.1 Statement of Extract-and-Simulate Lemma

Our lemma is stated as follows.

Lemma 4 (Extract-and-Simulate Lemma). *Let* C *be a finite set. Let* $\{\Pi_i\}_{i\in C}$ *be orthogonal projectors on a Hilbert space* $\mathcal{H}$ *such that the measurement* $\{\Pi_i, I-\Pi_i\}$ *can be efficiently implemented. Let* $|\psi_{\mathsf{init}}\rangle \in \mathcal{H}$ *be a unit vector.*

Suppose that there are a subset $S \in C^2$ *and a QPT algorithm* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *that satisfies the following:*

1. *S consists of an overwhelming fraction of C^2, i.e.,* $\frac{|S|}{|C|^2} = 1 - \mathsf{negl}(\lambda)$.
2. *For all $i \in C$, there exists a classical string s_i such that*
$$\Pr\left[\mathcal{A}_0\left(i, \frac{\Pi_i|\psi_{\mathsf{init}}\rangle}{\|\Pi_i|\psi_{\mathsf{init}}\rangle\|}\right) = s_i\right] = 1.$$
3. *There exists a classical string s^* such that for any $(i,j) \in S$,*
$$\Pr\left[\mathcal{A}_1\left(i, j, s_i, s_j\right) = s^*\right] = 1.$$

Let $\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ *be an experiment that works as follows:*

- *Choose $i \leftarrow C$.*
- *Apply the measurement $\{\Pi_i, I-\Pi_i\}$ on $|\psi_{\mathsf{init}}\rangle$.*
  - *If the state is projected onto Π_i, the experiment outputs i, the classical string s^*, and the resulting state* $\frac{\Pi_i|\psi_{\mathsf{init}}\rangle}{\|\Pi_i|\psi_{\mathsf{init}}\rangle\|}$.[22]
  - *If the state is projected onto $I-\Pi_i$, the experiment outputs i, $\bot$, and the resulting state* $\frac{(I-\Pi_i)|\psi_{\mathsf{init}}\rangle}{|(I-\Pi_i)|\psi_{\mathsf{init}}\rangle|}$.

Then, there is a QPT algorithm $\mathcal{SE}$ such that for any noticeable ε,

$$\{\mathcal{SE}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)\}_\lambda \stackrel{s}{\approx}_\varepsilon \{\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)\}_\lambda.$$

[22] We stress that we do not assume that the experiment is efficient. Especially, it may be computationally hard to find s^* from $\frac{\Pi_i|\psi_{\mathsf{init}}\rangle}{\|\Pi_i|\psi_{\mathsf{init}}\rangle\|}$.

4.2 Proof of the Extract-and-Simulate Lemma

We prove the extract-and-simulate lemma (Lem. 4). Throughout this subsection, we use the notations defined in Lem. 4.

Structure of the Proof. The high-level structure of our proof is similar to those of the (ε-)zero-knowledge properties of protocols in [BS20, CCY21]. We first construct extractors $\mathsf{Ext}_{\mathsf{Sim,a}}$ and $\mathsf{Ext}_{\mathsf{Sim,na}}$ that work in the "aborting case" and "non-aborting case", respectively where we say that the experiment $\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ aborts if its second output is $\bot$. Then we consider a combined simulator $\mathsf{Ext}_{\mathsf{Sim,comb}}$ that randomly guesses if the experiment aborts, runs either of $\mathsf{Ext}_{\mathsf{Sim,a}}$ or $\mathsf{Ext}_{\mathsf{Sim,na}}$ that corresponds to the guessed case, and returns a failure symbol Fail if the guess turns out to be wrong. Then, $\mathsf{Ext}_{\mathsf{comb}}$ correctly works conditioned on that the output is not Fail, and it returns Fail with probability almost 1/2. By applying Watrous' rewinding lemma (Lem. 2) to $\mathsf{Ext}_{\mathsf{Sim,comb}}$, we can convert it to a full-fledged simulator.

Let $\mathsf{Exp}_{\mathsf{a}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ and $\mathsf{Exp}_{\mathsf{na}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ be the same as $\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ except that they output a failure symbol Fail in aborting and non-aborting case, respectively. That is, they work as follows where differences from $\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$ are marked by red underlines:

$\mathsf{Exp}_{\mathsf{a}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$:

- Choose $i \leftarrow C$.
- Apply the measurement $\{\Pi_i, I - \Pi_i\}$ on $|\psi_{\mathsf{init}}\rangle$.
  - If the state is projected onto Π_i, the experiment outputs Fail.
  - If the state is projected onto $I - \Pi_i$, the experiment outputs i, $\bot$, and the resulting state $\frac{(I-\Pi_i)|\psi_{\mathsf{init}}\rangle}{\|(I-\Pi_i)|\psi_{\mathsf{init}}\rangle\|}$.

$\mathsf{Exp}_{\mathsf{na}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)$:

- Choose $i \leftarrow C$.
- Apply the measurement $\{\Pi_i, I - \Pi_i\}$ on $|\psi_{\mathsf{init}}\rangle$.
  - If the state is projected onto Π_i, the experiment outputs i, the classical string s^*, and the resulting state $\frac{\Pi_i|\psi_{\mathsf{init}}\rangle}{\|\Pi_i|\psi_{\mathsf{init}}\rangle\|}$.
  - If the state is projected onto $I - \Pi_i$, the experiment outputs Fail.

We give simulation extractors for each of these experiments.

Lemma 5 (Extract-and-Simulate for the Aborting Case). *There is a QPT algorithm* $\mathsf{Ext}_{\mathsf{Sim,a}}$ *such that for any noticeable* ε*,*

$$\{\mathsf{Ext}_{\mathsf{Sim,a}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)\}_\lambda \equiv \{\mathsf{Exp}_{\mathsf{a}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)\}_\lambda.^{23}$$

Proof of Lem. 5. Since $\mathsf{Exp}_{\mathsf{a}}$ can be run efficiently (because it never outputs s^*), $\mathsf{Ext}_{\mathsf{Sim,a}}$ just needs to run $\mathsf{Exp}_{\mathsf{a}}$. □

Lemma 6 (Extract-and-Simulate for the Non-aborting Case). *There is a QPT algorithm* $\mathsf{Ext}_{\mathsf{Sim,na}}$ *such that for any noticeable* ε*,*

$$\{\mathsf{Ext}_{\mathsf{Sim,na}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)\}_\lambda \stackrel{s}{\approx}_\varepsilon \{\mathsf{Exp}_{\mathsf{na}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)\}_\lambda.$$

[23] $\mathsf{Ext}_{\mathsf{Sim,a}}$ does not need to take $1^{\varepsilon^{-1}}$ or $\mathcal{A}$ as part of its input, but we include them in input for notational convenience.

Since the proof of Lem. 6 is the most non-trivial technical part, we defer it to Sec. 4.4 after some preparations in Sec. 4.3.

Given Lem. 5 and Lem. 6, the rest of the proof of Lem. 4 is very similar to the corresponding part of the ε-zero-knowledge property of the protocols in [CCY21]. We give the full proof for completeness.

Let $\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}$ be an algorithm that works as follows:

$\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$:

1. Set $\varepsilon' := \frac{\varepsilon^2}{3600\log^4(\lambda)}$.
2. Choose $\mathsf{mode} \leftarrow \{\mathsf{a}, \mathsf{na}\}$.
3. Run and output $\mathsf{Ext}_{\mathsf{Sim},\mathsf{mode}}(1^\lambda, 1^{\varepsilon'^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$.

Lemma 7 ($\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}$ Simulates Exp with Probability almost 1/2). *Let $p^{\mathsf{suc}}_{\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$ be the probability that $\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$ does not return Fail, and let*

$$D_{\mathsf{ext},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$$

be a conditional distribution of $\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$, conditioned on that it does not return Fail. Then we have

$$\left|p^{\mathsf{suc}}_{\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle) - 1/2\right| \leq \varepsilon'/2 + \mathsf{negl}(\lambda). \tag{4}$$

Moreover, we have

$$\{D_{\mathsf{ext},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)\}_\lambda \stackrel{s}{\approx}_{4\varepsilon'} \{\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)\}_\lambda. \tag{5}$$

Proof. (sketch.) The intuition behind this proof is as follows. By Lemma 5 and 6, $\mathsf{Ext}_{\mathsf{Sim},\mathsf{a}}$ and $\mathsf{Ext}_{\mathsf{Sim},\mathsf{na}}$ almost simulate Exp conditioned on that Exp aborts and does not abort, respectively. Therefore, if we randomly guess if Exp aborts and runs either of $\mathsf{Ext}_{\mathsf{Sim},\mathsf{a}}$ or $\mathsf{Ext}_{\mathsf{Sim},\mathsf{na}}$ that successfully works for the guessed case, the output distribution is close to the real output distribution of Exp conditioned on that the guess is correct, which happens with probability almost 1/2.

A formal proof can be obtained based on the above intuition and is exactly the same as the proof of [CCY21, Lemma 5.5] except for notational adaptations. □

Then, we convert $\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}$ into a full-fledged simulator that does not return Fail by using Watrous' rewinding lemma (Lemma 2). Namely, we let Q be a quantum algorithm that takes $|\psi_{\mathsf{init}}\rangle$ as input and outputs $\mathsf{Ext}_{\mathsf{Sim},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$ where $b := 0$ if and only if it does not return Fail, $p_0 := \frac{1}{4}$, $q := \frac{1}{2}$, $\gamma := \varepsilon'$, and $T := 2\log(1/\varepsilon')$. Then it is easy to check that the conditions for Lemma 2 is satisfied by Eq. 4 in Lemma 7 (for sufficiently large λ). Then by using Lemma 2, we can see that $\mathsf{R}(1^T, \mathsf{Q}, |\psi_{\mathsf{init}}\rangle)$ runs in time $T\cdot|\mathsf{Q}| = \mathsf{poly}(\lambda)$ and its output (seen as a mixed state) has a trace distance bounded by $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)}$ from $D_{\mathsf{ext},\mathsf{comb}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$. Since we have $\gamma = \varepsilon' = \frac{\varepsilon^2}{3600\log^4(\lambda)} = 1/\mathsf{poly}(\lambda)$, we have $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)} < 30\sqrt{\gamma}\log^2(\lambda) = \frac{\varepsilon}{2}$ for sufficiently large λ where we used $\log(1/\gamma) = \log(\mathsf{poly}(\lambda)) = o(\log^2(\lambda))$. Thus, by combining the above and Eq. 5 in Lemma 7, if we define $\mathsf{Ext}_{\mathsf{Sim}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle) := \mathsf{R}(1^T, \mathsf{Q}, |\psi_{\mathsf{init}}\rangle)$, then we have

$$\{\mathsf{Ext}_{\mathsf{Sim}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)\}_\lambda \stackrel{s}{\approx}_{\frac{\varepsilon}{2}+4\varepsilon'} \{\mathsf{Exp}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle)\}_\lambda.$$

We can conclude the proof of Lem. 4 by noting that we have $\frac{\varepsilon}{2} + 4\varepsilon' < \varepsilon$ since we have $\varepsilon' = \frac{\varepsilon^2}{3600\log^4(\lambda)} < \frac{\varepsilon}{8}$.

4.3 Preparation for Proof of Lem. 6

For proving Lem. 6, we prepare the following three lemmas.

The first is a simple technical lemma that is a variant of [CCY21, Lemma 3.1].

Lemma 8 (Variant of [CCY21, Lemma 3.1]). *Let $|\phi_b\rangle = |\phi_{b,0}\rangle + |\phi_{b,1}\rangle$ be a normalized quantum state in a Hilbert space $\mathcal{H}$. Let F be a quantum algorithm that takes a state in $\mathcal{H}$ as input and outputs a quantum state (not necessarily in $\mathcal{H}$) or a classical failure symbol* Fail*. Suppose that we have*

$$\Pr\left[F\left(\frac{|\phi_{b,0}\rangle\langle\phi_{b,0}|}{\||\phi_{b,0}\rangle\|^2}\right) = \mathsf{Fail}\right] \geq 1-\gamma$$

for $b \in \{0,1\}$ and $\||\phi_{1,1}\rangle - |\phi_{1,0}\rangle\| \leq \delta$. Then for any distinguisher D, it holds that[24]

$$|\Pr[D(F(|\phi_0\rangle\langle\phi_0|)) = 1] - \Pr[D(F(|\phi_1\rangle\langle\phi_1|)) = 1]| \leq (12\gamma^{1/2} + 2\delta)^{1/2}.$$

We give the proof in Appx. B.2.

The second lemma is a variant of the gentle measurement lemma shown in [CCY21].

Lemma 9 ([CCY21, Lemma 3.2]). *Let $|\psi\rangle_{\mathbf{X}}$ be a (not necessarily normalized) state over register $\mathbf{X}$ and U be a unitary over registers $(\mathbf{X},\mathbf{Y},\mathbf{Z})$. Suppose that a measurement of register $\mathbf{Z}$ of $U|\psi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y},\mathbf{Z}}$ results in a deterministic value except for probability ν, i.e., there is z^* such that*

$$\|(I - |z^*\rangle\langle z^*|_{\mathbf{Z}})U|\psi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y},\mathbf{Z}}\|^2 \leq \nu.$$

If we let $R := (|0\rangle\langle 0|)_{\mathbf{Y},\mathbf{Z}}U^\dagger(|z^*\rangle\langle z^*|)_{\mathbf{Z}}U$, then we have

$$\||\psi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y},\mathbf{Z}} - R|\psi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y},\mathbf{Z}}\| \leq \sqrt{\nu}.$$

The third is a variant of [CCY21, Lemma 3.3].

Lemma 10 (A variant of [CCY21, Lemma 3.3]). *Let Π be a projection over a Hilbert space $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$. For any noticeable function $\delta = \delta(\lambda)$, there exists an orthogonal decomposition $(S_{<\delta}, S_{\geq\delta})$ of $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$ that satisfies the following:*

1. *(**$S_{<\delta}$ and $S_{\geq\delta}$ are invariant under Π and $(|0\rangle\langle 0|)_{\mathbf{Y}}$.**) For any $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<\delta}$, we have*

$$\Pi|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<\delta}, \qquad (I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<\delta}.$$

   Similarly, for any $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq\delta}$, we have

$$\Pi|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq\delta}, \qquad (I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq\delta}.$$

2. *(**Π succeeds with probability $< \delta$ and $\geq \delta$ in $S_{<\delta}$ and $S_{\geq\delta}$.**) For any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{<\delta}$ we have*

$$\|\Pi|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}}\|^2 < \delta.$$

   Similarly, for any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{\geq\delta}$ we have

$$\|\Pi|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}}\|^2 \geq \delta.$$

[24] In the previous version of this paper, we claimed that the bound was $4\gamma^{1/4} + \||\phi_{0,1}\rangle\langle\phi_{0,1}| - |\phi_{1,1}\rangle\langle\phi_{1,1}|\|_{tr}$, but there was a flaw in the proof in the case of $|\phi_{0,1}\rangle \neq |\phi_{1,1}\rangle$.

3. (***Unitary for amplification.***) *For any $T \in \mathbb{N}$, there exists a unitary $U_{\mathsf{amp},T}$ over $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}} \otimes \mathcal{H}_{\mathbf{B}} \otimes \mathcal{H}_{\mathbf{Anc}}$ where $\mathbf{B}$ is a register to store a qubit and $\mathbf{Anc}$ is a register to store ancillary qubits with the following properties:*

   (a) (***Mapped onto $\Pi(I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})$ when $\mathbf{B}$ contains 1.***) *For any quantum state $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in \mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$, we can write*

$$|1\rangle\langle 1|_{\mathbf{B}} U_{\mathsf{amp},T} |\psi\rangle_{\mathbf{X},\mathbf{Y}} |0\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{anc} |\psi'_{anc}\rangle_{\mathbf{X},\mathbf{Y}} |1\rangle_{\mathbf{B}} |anc\rangle_{\mathbf{Anc}}$$

   by using sub-normalized states $|\psi'_{anc}\rangle_{\mathbf{X},\mathbf{Y}}$ that are in the span of $\Pi(I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})$.

   (b) (***Amplification of success probability in $S_{\geq\delta}$.***) *For any noticeable function $\nu = \nu(\lambda)$, there is $T = \mathsf{poly}(\lambda)$ such that for any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} \in S_{\geq\delta}$, we have*

$$\| |1\rangle\langle 1|_{\mathbf{B}} U_{\mathsf{amp},T} |\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} |0\rangle_{\mathbf{B},\mathbf{Anc}} \|^2 \geq 1 - \nu.$$

   (c) (***$S_{<\delta}$ and $S_{\geq\delta}$ are invariant under $U_{\mathsf{amp},T}$.***) *For any quantum state $|\psi_{<\delta}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<\delta}$ and any b, anc, we can write*

$$U_{\mathsf{amp},T} |\psi_{<\delta}\rangle_{\mathbf{X},\mathbf{Y}} |b, anc\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{b',anc'} |\psi'_{<\delta,b',anc'}\rangle_{\mathbf{X},\mathbf{Y}} |b', anc'\rangle_{\mathbf{B},\mathbf{Anc}}$$

   by using sub-normalized states $\left|\psi'_{<\delta,b',anc'}\right\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<\delta}$.
   Similarly, for any quantum state $\left|\psi_{\geq\delta}\right\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq\delta}$ and any b, anc, we can write

$$U_{\mathsf{amp},T} \left|\psi_{\geq\delta}\right\rangle_{\mathbf{X},\mathbf{Y}} |b, anc\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{b',anc'} \left|\psi'_{\geq\delta,b',anc'}\right\rangle_{\mathbf{X},\mathbf{Y}} |b', anc'\rangle_{\mathbf{B},\mathbf{Anc}}$$

   by using sub-normalized states $\left|\psi'_{\geq\delta,b',anc'}\right\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq\delta}$.

4. (***Efficient Implementation of $U_{\mathsf{amp},T}$.***) *There exists a QPT algorithm Amp (whose description is independent of Π) that takes as input 1^T, a description of quantum circuit that perform a measurement $(\Pi, I_{\mathbf{X},\mathbf{Y}} - \Pi)$, and a state $|\psi\rangle_{\mathbf{X},\mathbf{Y},\mathbf{B},\mathbf{Anc}}$, and outputs $U_{\mathsf{amp},T} |\psi\rangle_{\mathbf{X},\mathbf{Y},\mathbf{B},\mathbf{Anc}}$. Moreover, Amp uses the measurement circuit for only implementing an oracle that apply unitary to write a measurement result in a designated register in $\mathbf{Anc}$, and it acts on $\mathbf{X}$ only through the oracle access.*

The only difference from [CCY21, Lemma 3.3] is Item 3a where we require that an output of $U_{\mathsf{amp},T}$ to be in the span of $\Pi(I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})$ when $\mathbf{B}$ contains 1 whereas [CCY21, Lemma 3.3] only requires it to be in the span of Π.[25] We can prove Lem. 10 by using Jordan's lemma in a very similar manner to that for the proof of [CCY21, Lemma 3.3]. Thus, we defer the proof to Appx. B.1.

4.4 Proof of Lem. 6

By using Lem. 3 and 8 to 10, we prove Lem. 6, which completes the proof of Lem. 4.

Proof of Lem. 6. First, we define a "simulation-less extractor" $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$ that works as follows:
$\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi\rangle)$:

[25] In the previous version of this paper, we only required the state to be in the span of Π similarly to [CCY21, Lemma 3.3]. However, we found that we needed the above stronger requirement due to a technical reason. In particular,this is used in the proof of Claim 4.

1. Uniformly choose $(i,j) \leftarrow C^2$ and immediately output Fail if $(i,j) \notin S$.
2. For $k \in \{i,j\}$ in the order of $k=j$, $k=i$,[26] do the following:
   (a) Perform the measurement $\{\Pi_k, I-\Pi_k\}$, and immediately output Fail if the state is projected onto $I-\Pi_k$. Otherwise, go to the next step with the residual state $\left|\psi'_k\right\rangle$. [27]
   (b) Run $s_i \leftarrow \mathcal{A}_0(k, \left|\psi'_k\right\rangle)$ in a non-destructive way, i.e., in a way such that the state $\left|\psi'_k\right\rangle$ is preserved. This is possible since the output of $\mathcal{A}_0$ is deterministic as required in Item 2 of Lem. 4.
3. Run $s \leftarrow \mathcal{A}_1(i,j,s_i,s_j)$ and output s.

Then, by combining Lem. 3 and Item 3 of Lem. 4, we can show the following claim.

Claim 2. *For any $|\psi\rangle \in \mathcal{H}$, $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi\rangle)$ outputs s^* whenever it does not output Fail and it holds that*

$$\Pr\left[\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi\rangle) = s^*\right] \geq \left(\sum_{i\in C} \frac{1}{|C|} \|\Pi_i |\psi\rangle\|^2\right)^3 - \left(1 - \frac{|S|}{|C|^2}\right).$$

Proof of Claim 2. The former part is clear from Item 3 of Lem. 4. For the latter part, we only have to lower bound the probability that $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$ does not return Fail. If we remove the condition for outputting Fail in the first step of $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$, it is clear that the probability to not output Fail is $\sum_{i,j\in C^2} \frac{1}{|C|^2}\|\Pi_i\Pi_j |\psi\rangle\|^2$, which is lower bounded by $\left(\sum_{i\in C} \frac{1}{|C|}\|\Pi_i |\psi\rangle\|^2\right)^3$ by Lem. 3. Moreover, it is easy to see that the probability to output Fail in the first step of $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$ is $\frac{|S|}{|C|^2}$. By union bound, the latter part of the claim follows. □

Our next step is to apply Lem. 10 with respect to a projection corresponding to the success of $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}$. Let $U_{\mathsf{Sim\text{-}less}}$ be the unitary that represents $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, \cdot)$. More precisely, we define $U_{\mathsf{Sim\text{-}less}}$ over registers the input register Inp, working register $\mathbf{W}$, and output register $\mathbf{Out}$ so that $\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, \cdot))$ can be described as follows:

$\mathsf{Ext}_{\mathsf{Sim\text{-}less}}(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, \cdot))$: It takes a quantum state $|\psi\rangle$ in the register Inp and initializes registers $\mathbf{W}$ and $\mathbf{Out}$ to be $|0\rangle_{\mathbf{W},\mathbf{Out}}$. Then it applies the unitary $U_{\mathsf{Sim\text{-}less}}$, measures the register $\mathbf{Out}$ in the standard basis to obtain s, and outputs s.

We define a projection Π over $(\mathsf{Inp}, \mathbf{W}, \mathbf{Out})$ as

$$\Pi := U^\dagger_{\mathsf{Sim\text{-}less}} \left(\sum_{s\neq \mathsf{Fail}} |s\rangle\langle s|\right)_{\mathbf{Out}} U_{\mathsf{Sim\text{-}less}}. \tag{6}$$

Then the following claim immediately follows from the former half of Claim 2.

Claim 3. *Given any state in the span of $\Pi(I_{\mathsf{Inp}} \otimes (|0\rangle\langle 0|)_{\mathbf{W},\mathbf{Out}})$, if we apply $U_{\mathsf{Sim\text{-}less}}$ and then measure register $\mathbf{Out}$, then the measurement outcome is always s^**

We apply Lem. 10 for the above Π where $\mathcal{H}_{\mathbf{X}} := \mathcal{H}_{\mathsf{Inp}}$, $\mathcal{H}_{\mathbf{Y}} := \mathcal{H}_{\mathbf{W}} \otimes \mathcal{H}_{\mathbf{Out}}$, $\delta := \left(\frac{\varepsilon}{5}\right)^{12} - \left(1 - \frac{|S|}{|C|^2}\right)$, and $T = \mathsf{poly}(\lambda)$ is chosen in such a way that Item 3b of Lem. 10 holds for $\nu := \frac{\varepsilon^4}{16}$. Then we have a decomposition $(S_{<\delta}, S_{\geq\delta})$ of $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$ and a unitary $U_{\mathsf{amp},T}$ over $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}} \otimes \mathcal{H}_{\mathbf{B}} \otimes \mathcal{H}_{\mathbf{Anc}}$ that satisfy

[26] Whichever order is fine for our purpose. We specify it just for completeness of the description of the algorithm.
[27] Note that $\left|\psi'_j\right\rangle = \frac{\Pi_j|\psi\rangle}{|\Pi_j|\psi\rangle|}$ and $\left|\psi'_i\right\rangle = \frac{\Pi_i\Pi_j|\psi\rangle}{|\Pi_i\Pi_j|\psi\rangle|}$

the requirements in Lem. 10. We denote by **Other** to mean the registers **W**, **Out**, **B**, and **Anc** for brevity. We construct the extractor $\mathsf{Ext}_{\mathsf{Sim,na}}$ for Lem. 6 as follows:

$\mathsf{Ext}_{\mathsf{Sim,na}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle)$:

1. Set $|\psi_{\mathsf{init}}\rangle$ in register Inp and initlialize register **Other** to be $|0\rangle$.
2. Apply $U_{\mathsf{amp},T}$ by using the algorithm Amp in Item 4 of Lem. 10.
3. Measure register **B** and let b be the outcome. If $b = 0$, output Fail and immediately halt. Otherwise, proceed to the next step.
4. Apply $U_{\mathsf{Sim\text{-}less}}$, measure register **Out** to obtain an outcome s_{Ext}, and apply $U^\dagger_{\mathsf{Sim\text{-}less}}$.
5. Apply $U^\dagger_{\mathsf{amp},T}$ by using the algorithm Amp in Item 4 of Lem. 10.
6. Measure register **Other**. If the outcome is not the all 0's string, output Fail and immediately halt. Otherwise, let $|\psi_{\mathsf{mid}}\rangle$ be the state in register Inp at this point, and proceed to the next step.
7. Choose $i \leftarrow C$.
8. Apply the measurement $\{\Pi_i, I - \Pi_i\}$ on $|\psi_{\mathsf{mid}}\rangle$.
   - If the state is projected onto Π_i, output i, the classical string s_{Ext} and the resulting state $\frac{\Pi_i|\psi_{\mathsf{mid}}\rangle}{|\Pi_i|\psi_{\mathsf{mid}}\rangle|}$.
   - If the state is projected onto $I - \Pi_i$, output Fail.

We can easily see the following claim:

Claim 4. *Whenever Step 4 of* $\mathsf{Ext}_{\mathsf{Sim,na}}$ *is invoked,* s_{Ext} *obtained in the step is always equal to* s^**. Moreover, the step does not change the state in registers* Inp *and* **Other***, that is, the states before and after the step are identical.*

Proof of Claim 4. Whenever Step 4 is invoked, the bit b obtained in Step 3 is equal to 1. In this case, by Item 3a of Lem. 10, the state in registers Inp, **W**, and **Out** is in the span of $\Pi(I_{\mathsf{Inp}} \otimes (|0\rangle\langle 0|)_{\mathbf{W},\mathbf{Out}})$. Then, Claim 3 implies that s_{Ext} is always equal to s^*. Then the measurement of **Out** does not collapse the state and thus the step does not change the state. □

The rest of the proof is similar to that of [CCY21, Claim 4.5]. Let R be an operator defined as follows:

$$R := (|0\rangle\langle 0|)_{\mathbf{Other}} U^\dagger_{\mathsf{amp},T} (|1\rangle\langle 1|)_{\mathbf{B}} U_{\mathsf{amp},T}.$$

Let $\Pi_{<\delta}$ and $\Pi_{\geq\delta}$ be projections onto $S_{<\delta}$ and $S_{\geq\delta}$, respectively. To apply Lemma 8, we define states $|\phi_0\rangle = |\phi_{0,0}\rangle + |\phi_{0,1}\rangle$ and $|\phi_1\rangle = |\phi_{1,0}\rangle + |\phi_{1,1}\rangle$ over $(\mathbf{D}, \mathsf{Inp}, \mathbf{Other})$ where **D** is an additional one-qubit register as follows:

$$\begin{aligned}
|\phi_0\rangle &:= |1\rangle_{\mathbf{D}} |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{0,0}\rangle &:= |1\rangle_{\mathbf{D}} \Pi_{<\delta} |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{0,1}\rangle &:= |1\rangle_{\mathbf{D}} \Pi_{\geq\delta} |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}, \\
|\phi_1\rangle &:= |1\rangle_{\mathbf{D}} R |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}} + \alpha |0\rangle_{\mathbf{D}} |0\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{1,0}\rangle &:= |1\rangle_{\mathbf{D}} R\Pi_{<\delta} |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}} + \alpha |0\rangle_{\mathbf{D}} |0\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{1,1}\rangle &:= |1\rangle_{\mathbf{D}} R\Pi_{\geq\delta} |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}}
\end{aligned}$$

for $\alpha := \sqrt{1 - \| R |\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}} |0\rangle_{\mathbf{Other}} \|^2}$ (so that $|\phi_1\rangle$ is a normalized state). Let F be a quantum algorithm that works as follows:

$F\left(|\phi\rangle_{\mathbf{D},\mathsf{Inp},\mathbf{Other}}\right)$: It measures $\mathbf{D}$, and outputs Fail if the outcome is 0. Otherwise, it samples $i \leftarrow C$ and applies the measurement $\{\Pi_i, I-\Pi_i\}$ on register Inp. If the state is projected onto Π_i, output i, the measurement outcome s of the second register, and the resulting state in the third register. Otherwise, it outputs Fail.

It is easy to see that

$$\mathsf{Exp}_{\mathsf{na}}(\lambda, \{\Pi_i\}_{i\in C}, |\psi_{\mathsf{init}}\rangle) \equiv F(|\phi_0\rangle\langle\phi_0|).$$

Moreover, by the definition of $\mathsf{Ext}_{\mathsf{Sim,na}}$ and Claim 4, we can see that

$$\mathsf{Ext}_{\mathsf{Sim,na}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_i\}_{i\in C}, \mathcal{A}, |\psi_{\mathsf{init}}\rangle) \equiv F(|\phi_1\rangle\langle\phi_1|).$$

Thus, it suffices to prove that the distinguishing advantage between $F(|\phi_0\rangle\langle\phi_0|)$ and $F(|\phi_1\rangle\langle\phi_1|)$ is at most ε. To apply Lem. 8, we prove the following claim.

Claim 5. *The following hold:*

1. $\Pr\left[F\left(\frac{|\phi_{b,0}\rangle\langle\phi_{b,0}|}{\||\phi_{b,0}\rangle\|^2}\right) = \mathsf{Fail}\right] \geq 1-\left(\frac{\varepsilon}{5}\right)^4$ *for* $b\in\{0,1\}$.
2. $\| |\phi_{1,1}\rangle - |\phi_{0,1}\rangle \| \leq \left(\frac{\varepsilon}{2}\right)^2$.

Proof of Claim 5.
First item. We can write $\Pi_{<\delta}|\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}} = |\psi_{<\delta}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}}$. Then we have

$$\begin{aligned}
\Pr\left[F\left(\frac{|\phi_{0,0}\rangle\langle\phi_{0,0}|}{\||\phi_{0,0}\rangle\|^2}\right) \neq \mathsf{Fail}\right] &= \sum_{i\in C}\frac{1}{|C|}\left\|\Pi_i\frac{|\psi_{<\delta}\rangle}{\||\psi_{<\delta}\rangle\|}\right\|^2 \\
&\leq \left(\Pr\left[\mathsf{Ext}_{\mathsf{Sim\text{-}less}}\left(1^\lambda, \{\Pi_i\}_{i\in C}, \mathcal{A}, \frac{|\psi_{<\delta}\rangle}{\||\psi_{<\delta}\rangle\|}\right) = s^*\right] + \left(1-\frac{|S|}{|C|^2}\right)\right)^{1/3} \\
&\leq \left(\delta + \left(1-\frac{|S|}{|C|^2}\right)\right)^{1/3} \\
&= \left(\frac{\varepsilon}{5}\right)^4
\end{aligned}$$

where the first inequality follows from Claim 2, the second inequality follows from $|\psi_{<\delta}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}} \in S_{<\delta}$ and Item 2 of Lem. 10, and the final equality follows from $\delta = \left(\frac{\varepsilon}{5}\right)^{12} - \left(1-\frac{|S|}{|C|^2}\right)$. This completes the proof of the first item for the case of $b=0$. The case of $b=1$ can be proven similarly noting that $R\Pi_{<\delta}|\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}} \in S_{<\delta}$ by Item 1 and item 3c of Lemma 10.

Second Item. By Item 3b of Lem. 10, we have

$$\|(|1\rangle\langle 1|)_{\mathbf{B}}U_{\mathsf{amp},T}\Pi_{\geq t}|\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}}\|^2 \leq \nu.$$

Thus, Lem. 9 implies

$$\|\Pi_{\geq t}|\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}} - R\Pi_{\geq t}|\psi_{\mathsf{init}}\rangle_{\mathsf{Inp}}|0\rangle_{\mathbf{Other}}\| \leq \nu^{1/2}.$$

Since $\nu = \frac{\varepsilon^4}{16}$, this immediately implies the second item of the claim. □

By Lem. 8 and Claim 5 the distinguishing advantage between $F(|\phi_0\rangle\langle\phi_0|)$ and $F(|\phi_1\rangle\langle\phi_1|)$ is at most $\left(12\left(\frac{\varepsilon}{5}\right)^2 + 2\left(\frac{\varepsilon}{2}\right)^2\right)^{1/2} < \varepsilon$. This completes the proof of Lem. 6. □

5 Black-Box ε-Simulation-Extractable Commitments in Constant Rounds

In this section, we construct a post-quantum commitment scheme that satisfies the (parallel) strong extractability with ε-simulation. Namely, we prove the following lemma.

Lemma 11. *Assume the existence of post-quantum secure OWFs. Then, there exists a constant-round construction of post-quantum commitment that satisfies computational hiding (Def. 8), statistical binding (Def. 9), and (parallel) strongly extractable commitment with ε-simulation (Def. 11 and 12). Moreover, this construction makes only black-box use of the assumed OWF.*

Toward proving that, we first construct a scheme that satisfies a weaker notion of ε-simulatable extractability in Sec. 5.1. In Sec. 5.2, we present a compiler that converts the weak scheme in Sec. 5.1 into one that satisfies the (parallel) strong extractability with ε-simulation.

5.1 Weakly Extractable Commitment

We construct a commitment scheme that satisfies weak notions of extractability defined in Def. 13 and 16 based on OWFs. The description of the scheme is given in Prot. 1, where Com is a statistically-binding and computationally-hiding commitment scheme (e.g., Naor's commnitment). We remark that the scheme is identical to the *classical* extractable commitment in [PW09], which in turn is based on earlier works [DDN00, PRS02, Ros04].

Protocol 1: Extractable Commitment Scheme $\mathsf{wExtCom}$

The extractable commitment scheme, based on any commitment scheme Com, works in the following way.

Input:

- both the committer C and the receiver R get security parameter 1^λ as the common input.
- C gets a string $m \in \{0,1\}^{\ell(\lambda)}$ as his private input, where $\ell(\cdot)$ is a polynomial

Commitmment Phase:

1. The committer C commits using Com to $k = \lambda$ pairs of strings $\{(v_i^0, v_i^1)\}_{i=1}^k$ where $(v_i^0, v_i^1) = (\eta_i, m \oplus \eta_i)$ and η_i are random strings in $\{0,1\}^\ell$ for $1 \le i \le k$.[28] We denote those commitments by $\overline{\mathsf{com}} = \{\mathsf{com}_i^0, \mathsf{com}_i^1\}_{i=1}^k$.
2. Upon receiving a challenge $\mathbf{c} = (c_1, \ldots, c_k)$ from the receiver R, S opens the commitments to $\mathbf{v} := (v_1^{c_1}, \ldots, v_k^{c_k})$ with the corresponding decommitment $\overline{\mathsf{decom}} := (\mathsf{decom}_1^{c_1}, \ldots, \mathsf{decom}_k^{c_k})$.
3. R checks that the openings are valid.

Decommitment Phase:

- C sends σ and opens the commitments to all k pairs of strings. R checks that all the openings are valid, and also that $m = v_1^0 \oplus v_1^1 = \cdots = v_k^0 \oplus v_k^1$.

Proof of Security. The correctness and the statistically-binding property of $\mathsf{wExtCom}$ follows straightforwardly from that of Com. The computationally-hiding property of $\mathsf{wExtCom}$ can be reduced to that of Com by standard arguments.

Lemma 12 (Computational Hiding). $\mathsf{wExtCom}$ *is computationally hiding.*

Proof (sketch). Since this can be proven similarly to the classical counterpart in [PW09], we only give a proof sketch. For messages m_0, m_1 and $j \in [k+1]$, we consider a hybrid Hyb_j where (v_i^0, v_i^1) are 2-out-of-2 secret shares of m_0 for $i \ge j$ and those of m_1 for $i \le j-1$. What we should show is that Hyb_0 and Hyb_{k+1} are computationally indistinguishable from the view of a malicious receiver. By a standard hybrid argument, it suffices to prove the computational indistinguishability between Hyb_j and Hyb_{j+1}. This can be reduced to the computational hiding property of Com by guessing c_j and embedding the

[28] Actually, the scheme will be secure as long as we use Com to commit $k = \omega(\log \lambda)$ pairs of strings.

instance of computational hiding of Com into $\mathsf{com}_j^{1-c_j}$. The reduction works as long as the guess is correct, which occurs with probability 1/2. □

We prove that $\mathsf{wExtCom}$ satisfies a weak version of the extractability which we call the *weak extractability with ε-simulation*. Intuitively, it requires the simulation-extractor to perform extraction *and* ε-simulation properly, *as long as the commitment is valid*. A formal definition is given below.

Definition 13 (Weak Extractability with ε-Simulation). *A commitment scheme* Π *is* weakly extractable with ε-simulation *if there exists a QPT algorithm* $\mathcal{SE}_{\mathsf{weak}}$ *(called the ε-simulation weak-extractor) such that for any noticeable $\varepsilon(\lambda)$ and any non-uniform QPT $C^*(\rho)$,*

$$\begin{aligned}&\left\{\Gamma_{\mathsf{com}}(m_{\mathsf{Ext}},\widetilde{\mathsf{ST}}_{C^*}) : (\mathsf{com},m_{\mathsf{Ext}},\widetilde{\mathsf{ST}}_{C^*})\leftarrow \mathcal{SE}^{C^*(\rho)}(1^\lambda,1^{\varepsilon^{-1}})\right\}_\lambda\\ \overset{c}{\approx}_\varepsilon&\left\{\Gamma_{\mathsf{com}}(\mathsf{val}_\Pi(\mathsf{com}),\mathsf{ST}_{C^*}) : (\mathsf{com},\mathsf{ST}_{C^*},b_{\mathrm{com}})\leftarrow \langle C^*(\rho),R\rangle(1^\lambda)\right\}_\lambda\end{aligned}$$

where $\Gamma_{\mathsf{com}}(m,\mathsf{ST}_{C^*}) := \begin{cases}(m,\mathsf{ST}_{C^*}) & \text{if } \mathsf{val}_\Pi(\mathsf{com})\neq\bot\\ \bot & \text{otherwise}\end{cases}$.

Lemma 13 (Weak Extractability with ε-Simulation). $\mathsf{wExtCom}$ *is weakly extractable with ε-simulation (as per Def. 13).*

Before proving Lem. 13, we prepare several definitions.

Definition 14 (Validness of $\overline{\mathsf{com}}$). *For a sequence $\overline{\mathsf{com}}=\{\mathsf{com}_i^0,\mathsf{com}_i^1\}_{i=1}^k$ of commitments of the scheme Com, we say that $\overline{\mathsf{com}}$ is valid if there exists $m\in\{0,1\}^\ell$ such that $\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_i^b)\neq\bot$ for all $i\in[k]$ and $b\in\{0,1\}$ and $\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_i^0)\oplus\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_i^1)=m$ for all $i\in[k]$ where $\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_i^b)$ is the value function as defined in Def. 10. We denote by $\mathsf{val}_{\mathsf{Com}}(\overline{\mathsf{com}})$ to mean such m if $\overline{\mathsf{com}}$ is valid and otherwise $\bot$.*

Definition 15 (Accepting Opening of $\overline{\mathsf{com}}$). *For a sequence $\overline{\mathsf{com}}=\{\mathsf{com}_i^0,\mathsf{com}_i^1\}_{i=1}^k$ of commitments of the commitment scheme Com and $\mathbf{c}=(c_1,...,c_k)\in\{0,1\}^k$, we say that $(\mathbf{v}=(v_1,...,v_k),\overline{\mathsf{decom}}=(\mathsf{decom}_1,...,\mathsf{decom}_k))$ is an accepting opening of $\overline{\mathsf{com}}$ w.r.t. $\mathbf{c}$ if $\mathsf{Verify}_{\mathsf{Com}}(\mathsf{com}_i^{c_i},v_i,\mathsf{decom}_i)=1$ for all $i\in[k]$.*

Then we prove Lem. 13.

Proof of Lem. 13. For simplicity, we assume that Com satisfies perfect binding. It is straightforward to extend the proof to the statistically binding case by excluding the bad case where any commitment of Com is not bounded to a unique message, which happens with a negligible probability.

Remark that the weak extractability with ε-simulation only requires the extractor to correctly extract and simulate if the commitment generated in the commit stage is valid in the sense of Def. 10. When the commitment is valid, $\overline{\mathsf{com}}$ generated in Step 1 is also valid in the sense of Def. 14 (because otherwise a committer cannot pass the verification in the decommitment stage). Therefore, it suffices to prove that the extractor works for any fixed valid $\overline{\mathsf{com}}$.

Let $C^*(\rho)$ be a non-uniform QPT malicious committer. For $\mathbf{c}\in\{0,1\}^k$, let $U_{\mathbf{c}}$ be the unitary corresponding to the action of C^* in Step 2. That is, for the state ρ' before Step 2, it applies $U_{\mathbf{c}}$ to get $U_{\mathbf{c}}\rho' U_{\mathbf{c}}^\dagger$ and measures designated registers $\mathbf{V}$ and $\mathbf{D}$ to get the message $\mathbf{v}$ and opening information $\overline{\mathsf{decom}}$ in Step 2. Let $\Pi_{\mathbf{c}}^{\mathsf{test}}$ be the projection that maps onto states that contain an accepting opening $\mathbf{v}$ and $\overline{\mathsf{decom}}$ of $\overline{\mathsf{com}}$ w.r.t. $\mathbf{c}$ (as defined in Def. 15) in $\mathbf{V}\otimes\mathbf{D}$. For $\mathbf{c}\in\{0,1\}^k$, we define $\Pi_{\mathbf{c}}:=U_{\mathbf{c}}^\dagger\Pi_{\mathbf{c}}^{\mathsf{test}}U_{\mathbf{c}}$.

We apply Lem. 4 for $\{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}$ with the following correspondence.

– $\mathcal{H}$ is the internal space of C^*.

– The initial state is ρ'.[29]

– $C = \{0,1\}^k$.

– $S = \{((c_1,...,c_k),(c'_1,...,c'_k)) : \exists i \in [k] \text{ s.t. } c_i \neq c'_i\}$

– $\mathcal{A}_0$ applies $U_{\mathbf{c}}$ on its input, measures $\mathbf{V}$ to get $\mathbf{v}$, applies $U_{\mathbf{c}}^\dagger$, and outputs $\mathbf{v}$.

– $\mathcal{A}_1$ is given as input $(\mathbf{c},\mathbf{c}') \in S$, $\mathbf{v_c} = (v_1^{c_1},...,v_k^{c_k})$, and $\mathbf{v_{c'}} = (v_1^{c'_1},...,v_k^{c'_k})$. $\mathcal{A}_1$ outputs $v_i^{c_i} \oplus v_i^{c'_i}$ for the smallest $i \in [k]$ such that $c_i \neq c'_i$. Note that such i exists since we assume $(\mathbf{c},\mathbf{c}') \in S$.

If $\overline{\mathsf{com}}$ is valid, we can see that the assumptions for Lem. 4 are satisfied as follows:

1. By the definition of S, it is easy to see that $\frac{|S|}{|C|^2} = 1 - 2^{-k} = 1 - \mathsf{negl}(\lambda)$.
2. For any $\mathbf{c}$, if $\mathcal{A}_0$ takes a state in the span of $\Pi_{\mathbf{c}}$ as input, it outputs $s_{\mathbf{c}} := (\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_1^{c_1}),...,\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_k^{c_k}))$ with probability 1 by the definition of $\Pi_{\mathbf{c}}$ and the perfect binding property of Com.
3. For any $(\mathbf{c},\mathbf{c}') \in S$, if $\mathcal{A}_1$ takes as input the $s_{\mathbf{c}}$ and $s_{\mathbf{c}'}$ defined as follows:

$$\begin{cases} s_{\mathbf{c}} = (\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_1^{c_1}),\dots,\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_k^{c_k})) \\ s_{\mathbf{c}'} = (\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_1^{c'_1}),\dots,\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_k^{c'_k})) \end{cases};$$

then, it outputs $s^* := \mathsf{val}_{\mathsf{Com}}(\overline{\mathsf{com}})$ as defined in Def. 14 since we assume that $\overline{\mathsf{com}}$ is valid.

Let $\widetilde{\mathcal{SE}}$ be the ε-simulation extractor of Lem. 4 in the above setting. Then Lem. 4 gives us the following:

$$\{\widetilde{\mathcal{SE}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \mathcal{A}, \rho')\}_\lambda \stackrel{\mathrm{s}}{\approx}_\varepsilon \{\mathsf{Exp}(\lambda, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \rho')\}_\lambda$$

where $\mathsf{Exp}(\lambda, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \rho')$ is as defined in Lem. 4. That is, $\mathsf{Exp}(\lambda, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \rho')$ works as follows:

– Choose $\mathbf{c} \leftarrow \{0,1\}^k$.

– Apply the measurement $\{\Pi_{\mathbf{c}}, I - \Pi_{\mathbf{c}}\}$ on ρ'.

- If the state is projected onto $\Pi_{\mathbf{c}}$, the experiment outputs $\mathbf{c}$, the classical string $\mathsf{val}_{\mathsf{Com}}(\overline{\mathsf{com}})$, and the resulting state.
- If the state is projected onto $I - \Pi_{\mathbf{c}}$, the experiment outputs $\mathbf{c}$, $\bot$, and the resulting state.

One can see that the state in the third output of $\mathsf{Exp}(\lambda, \rho')$ is similar to the final state of C^* in the real execution except that C^* applies the unitary $U_{\mathbf{c}}$ instead of the measurement $\{\Pi_{\mathbf{c}}, I - \Pi_{\mathbf{c}}\}$ and measures $\mathbf{V}$ and $\mathbf{D}$. By noting that $\Pi_{\mathbf{c}}^{\mathsf{test}} U_{\mathbf{c}} = U_{\mathbf{c}} \Pi_{\mathbf{c}}$ and that measuring $\mathbf{V}$ and $\mathbf{D}$ is the same as first applying the measurement $\{\Pi_{\mathbf{c}}^{\mathsf{test}}, I - \Pi_{\mathbf{c}}^{\mathsf{test}}\}$ and then measuring $\mathbf{V}$ and $\mathbf{D}$, if we apply $U_{\mathbf{c}}$ on the third output of $\mathsf{Exp}(\lambda, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \rho')$ and then measure $\mathbf{V}$ and $\mathbf{D}$, the state is exactly the same as the final state of C^*.

Therefore, the following extractor $\mathcal{SE}_{\mathsf{weak}}$ works for the weak ε-simulation extractability:

$\mathcal{SE}_{\mathsf{weak}}^{C^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})$:

1. Run the commit stage of $\mathsf{wExtCom}$ between $C^*(\rho)$ and the honest receiver R until C^* sends $\overline{\mathsf{com}}$ in Step 1. Let ρ' be the internal state of C^* at this point.
2. Run $(\mathbf{c}, m_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \leftarrow \widetilde{\mathcal{SE}}(1^\lambda, 1^{\varepsilon^{-1}}, \{\Pi_{\mathbf{c}}\}_{\mathbf{c}\in\{0,1\}^k}, \mathcal{A}, \rho')$ where $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ is as defined above. Remark that the definition of $\Pi_{\mathbf{c}}$ depends on $\overline{\mathsf{com}}$, and it uses $\overline{\mathsf{com}}$ generated in the previous step.

[29] Though we assume that the initial state $|\psi_{\mathsf{init}}\rangle$ is a pure state in Lem. 4, the lemma holds for any mixed state since a mixed state can be seen as a probability distribution over pure states.

3. Apply $U_{\mathbf{c}}$ on ρ_{Ext} to generate $U_{\mathbf{c}}\rho_{\mathsf{Ext}}U_{\mathbf{c}}^{\dagger}$ and measures registers $\mathbf{V}$ and $\mathbf{D}$ to get $\mathbf{v}$ and $\overline{\mathsf{decom}}$. Let ρ_{final} be the state after the measurement.
4. Output $(m_{\mathsf{Ext}}, \rho_{\mathsf{final}})$.

□

On the Parallel Execution of wExtCom**.** We can prove that wExtCom satisfies a parallel version of the weak extractability with ε-simulation in a similar way. In the following, we prove that wExtCom satisfies even a generalized version of that, which we call *special parallel weak extractability with ε-simulation*. Looking ahead, this will be used in the proof of the (parallel) ε-simulation strong extractability of Prot. 2 in Sec. 5.2.

Intuitively, it requires the following: Suppose that a malicious committer C^* interacts with n copies of the honest receiver R in parallel, and let com_j be the commitment generated in the j-th execution. Suppose that com_j is valid for all $j \in V$ for some subset $V \subseteq [n]$. Let $F : \{0,1\}^{\ell} \cup \{\bot\} \to \{0,1\}^*$ be a function that is determined by $\{\mathsf{val}(\mathsf{com}_j)\}_{j\in V}$, i.e., $F(m_1, ..., m_n)$ takes a unique value m^* as long as $m_j = \mathsf{val}(\mathsf{com}_j)$ for all $j \in V$. Then, the extractor can extract m^* while simulating the post-execution state of C^*. A formal definition is given below.

Definition 16 (Special Parallel Weak Extractability with ε-Simulation). *We say that a commitment scheme* Π *satisfies the* special parallel weak extractability with ε-simulation *if the following is satisfied. For any integer $n = \mathsf{poly}(\lambda)$ and an efficiently computable function $F : \{\{0,1\}^{\ell} \cup \{\bot\}\}^n \to \{0,1\}^*$, there exists $\mathcal{SE}_F$ that satisfies the following: For commitments $\{\mathsf{com}_j\}_{j=1}^n$, we say that $\{\mathsf{com}_j\}_{j=1}^n$ is F-good if it satisfies the following:*

1. *there exists $V \subseteq [n]$ such that com_j is valid (i.e., $\mathsf{val}_{\Pi}(\mathsf{com}_j) \neq \bot$) for all $j \in V$;* **and**
2. *there exists a unique m^* such that $F(m'_1, ..., m'_n) = m^*$ for all $(m'_1, ..., m'_n)$ such that $m'_j = \mathsf{val}_{\Pi}(\mathsf{com}_j)$ for all $j \in V$.*

Then it holds that

$$\left\{\Gamma_{F,\{\mathsf{com}_j\}_{j=1}^n}(m_{\mathsf{Ext}}, \mathsf{ST}_{C^*}) : (\{\mathsf{com}_j\}_{j=1}^n, m_{\mathsf{Ext}}, \mathsf{ST}_{C^*}) \leftarrow \mathcal{SE}_F^{C^*(\rho)}(1^{\lambda}, 1^{\varepsilon^{-1}})\right\}_{\lambda}$$
$$\overset{c}{\approx}_{\varepsilon} \left\{\begin{array}{l}\Gamma_{F,\{\mathsf{com}_j\}_{j=1}^n}(F(\mathsf{val}_{\Pi}(\mathsf{com}_1), ..., \mathsf{val}_{\Pi}(\mathsf{com}_n)), \mathsf{ST}_{C^*}) \\ : (\{\mathsf{com}_j\}_{j=1}^n, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^n) \leftarrow \langle C^*(\rho), R^n\rangle(1^{\lambda})\end{array}\right\}_{\lambda},$$

where $(\{\mathsf{com}_j\}_{j=1}^n, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^n) \leftarrow \langle C^*(\rho), R^n\rangle(1^{\lambda})$ means that $C^*(\rho)$ interacts with n copies of the honest receiver R in parallel and the execution results in transcripts $\{\mathsf{com}_j\}_{j=1}^n$, the final state ST_{C^*}, and outputs $\{b_{\mathrm{com},j}\}_{j=1}^n$ of each copy of R and

$$\Gamma_{F,\{\mathsf{com}_j\}_{j=1}^n}(m, \mathsf{ST}_{C^*}) := \begin{cases}(m, \mathsf{ST}_{C^*}) & \text{if } \{\mathsf{com}_j\}_{j=1}^n \text{ is } F\text{-good} \\ \bot & \text{otherwise}\end{cases}.$$

Lemma 14 (Special Parallel Weak Extractability with ε-Simulation). wExtCom *satisfies the special parallel weak extractability with ε-simulation (as per Def. 16).*

Proof. Since the proof is very similar to that of Lem. 13, we only highlight the differences from that. Similarly to the proof of Lem. 13, we assume that Com satisfies perfect binding for simplicity.

In a parallel interaction between a non-uniform QPT malicious committer $C^*(\rho)$ and n copies of R, let $\mathsf{wExtCom.com}_j$ be the commitment generated in the j-th session and let $\overline{\mathsf{com}}_j = \{\mathsf{com}_{j,i}^0, \mathsf{com}_{j,i}^1\}_{i=1}^k$ be the part of $\mathsf{wExtCom.com}_j$ that consists of the commitments of Com generated in Step 1. Remark

that the special parallel weak extractability with ε-simulation only requires the extractor to correctly extract and simulate if $\{\mathsf{wExtCom.com}_j\}_{j=1}^n$ is F-good. In this case, $\{\overline{\mathsf{com}}_j\}_{j=1}^n$ is also F-good in the following sense:

- there exists $V \subseteq [n]$ such that $\overline{\mathsf{com}}_j$ is valid (as per Def. 14) for all $j \in V$, and
- there exists m^* such that $F(m'_1, ..., m'_n) = m^*$ for all $(m'_1, ..., m'_n)$ such that $m'_j = \mathsf{val}_{\mathsf{Com}}(\overline{\mathsf{com}}_j)$ for all $j \in V$.

Therefore, it suffices to prove that the extractor works for any fixed F-good $\{\overline{\mathsf{com}}_j\}_{j=1}^n$.

For $\{\mathbf{c}_j\}_{j=1}^n \in (\{0,1\}^k)^n$, let $U_{\{\mathbf{c}_j\}_{j=1}^n}$ be the unitary corresponding to the action of C^* in Step 2 similarly to the proof of Lem. 13. Remark that the unitary is indexed by $\{\mathbf{c}_j\}_{j=1}^n$ since we are considering an n-parallell execution. Similarly, we let $\Pi^{\mathsf{test}}_{\{\mathbf{c}_j\}_{j=1}^n}$ be the projection that maps onto states that contain accepting openings of $\overline{\mathsf{com}}_j$ w.r.t. $\mathbf{c}_j$ for all $j \in [n]$ in the designated registers. Then, we define $\Pi_{\{\mathbf{c}_j\}_{j=1}^n} := U^\dagger_{\{\mathbf{c}_j\}_{j=1}^n} \Pi^{\mathsf{test}}_{\{\mathbf{c}_j\}_{j=1}^n} U_{\{\mathbf{c}_j\}_{j=1}^n}$.

We apply Lem. 4 for $\{\Pi_{\{\mathbf{c}_j\}_{j=1}^n}\}_{\{\mathbf{c}_j\}_{j=1}^n \in (\{0,1\}^k)^n}$ with the following correspondence.

- $\mathcal{H}$ is the internal space of C^*.
- The initial state is ρ'.
- $C = (\{0,1\}^k)^n$.
- $S = \{(\{c_{j,1}, ..., c_{j,k})\}_{j=1}^n, \{c'_{j,1}, ..., c'_{j,k}\}_{j=1}^n) : \forall j \in [n]\ \exists i \in [k] \text{ s.t. } c_{j,i} \neq c'_{j,i}\}$
- $\mathcal{A}_0$ applies $U_{\{\mathbf{c}_j\}_{j=1}^n}$ on its input, measures the designated registers to get the $\mathbf{v}_j$ for each $j \in [n]$, applies $U^\dagger_{\{\mathbf{c}_j\}_{j=1}^n}$, and outputs $\{\mathbf{v}_j\}_{j=1}^n$.
- $\mathcal{A}_1$ is given as input $(\{\mathbf{c}_j\}_{j=1}^n, \{\mathbf{c}'_j\}_{j=1}^n) \in S$, $\mathbf{v}_{\{\mathbf{c}_j\}_{j=1}^n} = \{v_{j,1}^{c_{j,1}}, ..., v_{j,k}^{c_{j,k}}\}_{j=1}^n$, and $\mathbf{v}_{\{\mathbf{c}'_j\}_{j=1}^n} = \{v_{j,1}^{c'_{j,1}}, ..., v_{j,k}^{c'_{j,k}}\}_{j=1}^n$. $\mathcal{A}_1$ computes $m_j := v_{j,i_j}^{c_{j,i_j}} \oplus v_{j,i_j}^{c'_{j,i_j}}$ for $j \in [n]$ where $i_j \in [k]$ is the smallest index such that $c_{j,i} \neq c'_{j,i}$ and outputs $F(m_1, ..., m_n)$. Note that such i_j exists for all $j \in [n]$ since we assume $(\{\mathbf{c}_j\}_{j=1}^n, \{\mathbf{c}'_j\}_{j=1}^n) \in S$.

If $\{\overline{\mathsf{com}}_j\}_{j=1}^n$ is F-good, we can see that the assumptions for Lem. 4 are satisfied as follows.

1. By the definition of S and the union bound, it is easy to see that $\frac{|S|}{|C|^2} \geq 1 - n2^{-k} = 1 - \mathsf{negl}(\lambda)$.
2. For any $\{\mathbf{c}_j\}_{j=1}^n$, if $\mathcal{A}_0$ takes a state in the span of $\Pi_{\{\mathbf{c}\}_{j=1}^n}$ as input, it outputs $s_{\{\mathbf{c}_j\}_{j=1}^n} := \{\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,1}^{c_{j,1}}), ..., \mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,k}^{c_{j,k}})\}_{j=1}^n$ with probability 1 by the definition of $\Pi_{\{\mathbf{c}\}_{j=1}^n}$ and the perfect binding property of Com.
3. Let $V \subseteq [n]$ be a subset for which the conditions of the F-goodness are satisfied. For any $(\{\mathbf{c}_j\}_{j=1}^n, \{\mathbf{c}'_j\}_{j=1}^n) \in S$, if $\mathcal{A}_1$ takes as input the $s_{\{\mathbf{c}_j\}_{j=1}^n}$ and $s_{\{\mathbf{c}'_j\}_{j=1}^n}$ defined as follows:

$$\begin{cases} s_{\{\mathbf{c}_j\}_{j=1}^n} := \{\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,1}^{c_{j,1}}), ..., \mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,k}^{c_{j,k}})\}_{j=1}^n \\ s_{\{\mathbf{c}'_j\}_{j=1}^n} := \{\mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,1}^{c'_{j,1}}), ..., \mathsf{val}_{\mathsf{Com}}(\mathsf{com}_{j,k}^{c'_{j,k}})\}_{j=1}^n \end{cases};$$

then, we have $m_j = \mathsf{val}_{\mathsf{Com}}(\overline{\mathsf{com}}_j)$ for all $j \in V$ since $\overline{\mathsf{com}}_j$ is valid for all $j \in V$. Note that this may not hold for $j \notin V$. However, by the second condition of the F-goodness, there is m^* such that $F(m_1, ..., m_n) = m^*$ regardless of the values of $\{m_j\}_{j \notin V}$. We can set $s^* := m^*$.

Then, the rest of the proof is identical to that of Lem. 13. □

5.2 Strongly Extractable Commitment

In this section, we construct *strongly* extractable commitment with ε-simulation. The scheme is shown in Prot. 2. It relies on the following building blocks:

1. the ε-simulatable *weakly* extractable commitment $\mathsf{wExtCom}$ given in Prot. 1. We remark that the security of Prot. 2 relies on the particular $\mathsf{wExtCom}$ presented in Prot. 1 because we also need the special parallel weak extractability with ε-simulation (Def. 16); we do not know if Prot. 2 can be based on any $\mathsf{wExtCom}$ satisfying the weak extractability with ε-simulation as in Def. 13.
2. a $(n+1,t)$-perfectly verifiable secret sharing scheme $\mathsf{VSS} = (\mathsf{VSS}_{\mathsf{Share}}, \mathsf{VSS}_{\mathsf{Recon}})$ (as per Def. 1). We require that t is a constant fraction of n such that $t \leq n/3$. There are known constructions (without any computational assumptions) satisfying these properties [BGW88, CDD+99].

Protocol 2: ε-Simulatable Strongly Extractable Commitment ExtCom

Let $n(\lambda)$ be a polynomial on λ. Let t be a constant fraction of n such that $t \leq n/3$.

Input: both the (committer) C and the receiver R get security parameter 1^λ as the common input; C gets a string $m \in \{0,1\}^{\ell(\lambda)}$ as his private input, where $\ell(\cdot)$ is a polynomial.

Commit Stage:

1. C emulates $n+1$ (virtual) players $\{P_i\}_{i\in[n+1]}$ to execute the $\mathsf{VSS}_{\mathsf{Share}}$ protocol "in his head", where the input to P_{n+1} (i.e., the Dealer) is m. Let $\{\mathsf{v}_i\}_{i\in[n+1]}$ be the views of the $n+1$ players describing the execution.
2. C and R involve in n executions of $\mathsf{wExtCom}$ in parallel, where in the i-th instance ($i \in [n]$), C commits to v_i.
3. R picks a random string r_1 and commits to it using $\mathsf{wExtCom}$.
4. C picks a random string r_2 and sends it to R.
5. R sends to C the value r_1 together with the corresponding decommitment information w.r.t. the $\mathsf{wExtCom}$ in Step 3. Now, both parties learn a coin-tossing result $r = r_1 \oplus r_2$, which specifies a size-t random subset $T \subseteq [n]$.
6. C sends to R in *one round* the following messages: $\{\mathsf{v}_i\}_{i\in T}$ together with the corresponding decommitment information w.r.t. the $\mathsf{wExtCom}$ in Step 2.
7. R checks the following conditions:
   (a) All the decommitments in Step 6 are valid; **and**
   (b) for any $i, j \in T$, views $(\mathsf{v}_i, \mathsf{v}_j)$ are consistent (as per Def. 3 and Rmk. 2) w.r.t. the $\mathsf{VSS}_{\mathsf{Share}}$ execution as described in Step 1.

   If all the checks pass, R accepts; otherwise, R rejects.

Decommit Stage:

1. C sends $\{\mathsf{v}_i\}_{i\in[n]}$ together with all the corresponding information w.r.t. the $\mathsf{wExtCom}$ in Step 1 of the Commit Stage.
2. R constructs $\{\mathsf{v}'_i\}_{i\in[n]}$ as follows: in Step 1 of the Decommit Stage, if the i-th decommitment is valid, R sets $\mathsf{v}'_i := \mathsf{v}_i$; otherwise, R sets $\mathsf{v}'_i := \perp$.
3. R outputs $m' := \mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}'_1, \ldots, \mathsf{v}'_n)$.

Proof of Security. Correctness and statistically-binding property of ExtCom follows straightforwardly from that of $\mathsf{wExtCom}$. In the following, we prove that ExtCom is computationally-hiding and (parallel) strong extractable with ε-simulation.

Lemma 15 (Computational Hiding). ExtCom *is computationally hiding.*

Proof. We consider an interaction between the honest sender C and a malicious non-uniform QPT receiver $R^*(\rho)$. We consider the following two cases:

Case 1: If C rejects the decommitment in Step 5, any commitment of $\mathsf{wExtCom}$ generated in Step 2 is not decomitted. In this case, it is straightforward to reduce the computational hiding to that of $\mathsf{wExtCom}$.

Case 2: If C accepts the decommitment in Step 5, the commitment of $\mathsf{wExtCom}$ generated in Step 3 is valid (as per Def. 10). We consider the following hybrid experiments, where H_0 denotes the real execution $\langle C, R^*(\rho)\rangle(1^\lambda)$:

- Hybrid H_1: This hybrid works similarly to H_0 except that we run the ε-simulation weak extractor to extract r_1 from the commitment in Step 3. Since the commitment is valid in this case as observed above, the final output of R^* is ε-close to that in H_0 for arbitrarily small noticeable ε.
- Hybrid H_2: In the next hybrid, we first randomly pick a size-t random subset T at the beginning, and define r_2 so that $r = r_1 \oplus r_2$ specifies T in Step 4. Since r_2 is uniformly distributed in either case, this is perfectly indistinguishable from H_1.
- Hybrid H_3: This hybrid is identical to H_2, except that in Step 1, for all $i \in [n] \setminus T$, we set v_i to an all-0 string of proper length.
  $\underline{H_2 \overset{\text{c}}{\approx} H_3}$: We now reduce the indistinguishability between H_2 and H_3 to the computationally-hiding property of (parallel executions of)[30] $\mathsf{wExtCom}$. Consider an adversary $\mathcal{A}$ participating in the hiding game of $\mathsf{wExtCom}$. $\mathcal{A}$ finishes Step 1 as in H_2, where the views of the n parties are denoted as $\{\mathsf{v}_i\}_{i\in[n]}$. It also prepare $(n-t)$ "null views" $\{\mathsf{v}'_i\}_{i\in[n]\setminus T}$ where each of them is an all-0 string of proper length. $\mathcal{A}$ sends $\{\mathsf{v}_i\}_{i\in[n]\setminus T}$ and $\{\mathsf{v}'_i\}_{i\in[n]\setminus T}$ to the external challenger as his challenges for the parallel hiding game of $\mathsf{wExtCom}$. Then, $\mathcal{A}$ generates the Step 2 commitment (in parallel) in the following manner:
  - For $i \in T$: $\mathcal{A}$ commits to $\{\mathsf{v}_i\}_{i\in T}$ to (the internal) R^* using $\mathsf{wExtCom}$ himself;
  - For $i \in [n] \setminus T$: $\mathcal{A}$ relays the external challenger's $\mathsf{wExtCom}$ commitments to the internal R^*.

  Finally, $\mathcal{A}$ finishes the remaining steps of the Commit Stage as in H_3 (or equivalently, H_2). We remark that in Step 6, $\mathcal{A}$ needs to decommit to the views corresponding to T, which he can because these commitments are generated by $\mathcal{A}$ himself.
  Now, observe that if the external challenger commits to $\{\mathsf{v}_i\}_{i\in[n]\setminus T}$, then $\mathcal{A}$ is identical to H_2; if the external challenger commits to $\{\mathsf{v}'_i\}_{i\in[n]\setminus T}$, then $\mathcal{A}$ is identical to H_3. Therefore, $H_2 \overset{\text{c}}{\approx} H_3$ as otherwise $\mathcal{A}$ wins the parallel hiding game.
- Hybrid H_4: This hybrid is identical to H_3, except that in Step 1, we run the simulator guaranteed by the Secrecy property (Property 2) of VSS to fake the views of parties in set T: $\{\mathsf{v}'_i\}_{i\in[T]} \leftarrow \mathcal{S}^{\mathcal{A}}(1^\lambda, T)$, and for each $j \in [n] \setminus T$, we set v'_j to all-0 strings of proper length. Note that this hybrid does not need to know m anymore. It is easy to see that $H_3 \overset{\text{i.d.}}{=\!=} H_4$ due to the perfect secrecy of VSS.

The above argument implies that for any m_0 and m_1, the outputs of R^* when the message is m_0 or m_1 are indistinguishable with advantage at most 2ε. Since ε is arbitrarily small noticeable function, the above implies that they are computationally indistinguishable in the standard sense.

Since we know that one of Case 1 and Case 2 must happen, the overall reduction algorithm can first guess which case occurs and run the reduction algorithm corresponding to the guessed case; when the guess turns out to be incorrect, it simply outputs a uniform bit. Since the guess is correct with probability $1/2$, this reduction works with a security loss of the multiplicative factor $1/2$.[31] □

[30] Note that the computationally-hiding property of parallel executions follows from that of the stand-alone execution by a standard hybrid argument.

[31] If we use Watrous' rewinding lemma (Lem. 2), we may avoid the security loss; but this is not needed here.

In the following, we prove the (parallel-)strong extractability with ε-simulation. Though we finally prove the parallel version, we first give a proof for the stand-alone version since that is simpler and the proof is readily extended to that of the parallel version.

Lemma 16 (Strong Extractability with ε-Simulation). ExtCom *is strongly extractable with ε-simulation (as per Def. 11).*

Proof. Suppose that a non-uniform QPT committer C^* interacts with the honest receiver R in the commit stage of ExtCom. We consider two cases where R accepts or rejects, respectively. By using Watrous' rewinding lemma (Lem. 2) in a similar way to the proof of Lem. 4, it suffices to construct a simulator that correctly extracts and simulates for each case separately. Moreover, when R rejects, the commitment is invalid and thus the extractor does not need to extract anything. Thus, there is a trivial perfect simulation extractor for this case: it can simply run the interaction between $C^*(\rho)$ and R by playing the role of R and outputs the final state of C^*. What is left is to construct an extractor that correctly extracts and simulates assuming that R accepts in the committing stage. That is, it suffices to prove the following claim.

Claim 6 (Extraction and Simulation for Accepting Case)**.** *There exists a QPT algorithm $\mathcal{SE}_{\mathsf{Acc}}$ such that for any noticeable $\varepsilon(\lambda)$ and any non-uniform QPT $C^*(\rho)$, it holds that*

$$\begin{aligned}&\left\{\Gamma_{b_{\mathrm{com}}}(m_{\mathsf{Ext}},\mathsf{ST}_{C^*}) : (m_{\mathsf{Ext}},\mathsf{ST}_{C^*},b_{\mathrm{com}}) \leftarrow \mathcal{SE}_{\mathsf{Acc}}^{C^*(\rho)}(1^\lambda,1^{\varepsilon^{-1}})\right\}_\lambda\\ \overset{c}{\approx}_\varepsilon&\left\{\Gamma_{b_{\mathrm{com}}}(\mathsf{val}_{\mathsf{ExtCom}}(\mathsf{com}),\mathsf{ST}_{C^*}) : (\mathsf{com},\mathsf{ST}_{C^*},b_{\mathrm{com}}) \leftarrow \langle C^*(\rho),R\rangle(1^\lambda)\right\}_\lambda,\end{aligned}$$

where $\Gamma_{b_{\mathrm{com}}}(m,\mathsf{ST}_{C^*}) := \begin{cases}(m,\mathsf{ST}_{C^*}) & \text{if } b_{\mathrm{com}}=1\\ \bot & \text{otherwise}\end{cases}$.

Remark 5. One may think that the above claim is similar to the weak extractability with ε-simulation (Def. 13). However, the crucial difference is that the extractor $\mathcal{SE}_{\mathsf{Acc}}$ should declare if the simulation has succeeded by outputting b_{com} in the clear. On the other hand, in Def. 13, $\mathcal{SE}_{\mathsf{weak}}$ is only required to indirectly declare that depending on if com is valid, which may not be known by $\mathcal{SE}_{\mathsf{weak}}$.

Proof of Claim 6. Let $\mathsf{wExtCom.com}_i$ be the i-th commitment of $\mathsf{wExtCom}$ in Step 2 in the commit stage. In the execution of $(\mathsf{com},\mathsf{ST}_{C^*},b_{\mathrm{com}}) \leftarrow \langle C^*(\rho),R\rangle(1^\lambda)$, let Good be the event that $\{\mathsf{wExtCom.com}_i\}_{i=1}^n$ is $\mathsf{VSS}_{\mathsf{Recon}}$-good in the sense of Def. 16, i.e.,

- there exists $V \subseteq [n]$ such that $\mathsf{wExtCom.com}_i$ is valid (i.e., $\mathsf{val}_{\mathsf{wExtCom}}(\mathsf{wExtCom.com}_i) \neq \bot$) for all $i \in V$, and
- there exists m^* such that $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}'_1,\dots,\mathsf{v}'_n) = m^*$ for all $(\mathsf{v}'_1,\dots,\mathsf{v}'_n)$ such that

$$\forall i \in V,\ \mathsf{v}'_i = \mathsf{val}_{\mathsf{wExtCom}}(\mathsf{wExtCom.com}_i).$$

Let Bad be the complementary event of Good. We prove the following claim.

Claim 7. *It holds that*

$$\Pr\left[\mathsf{Bad} \wedge b_{\mathrm{com}}=1 : (\mathsf{com},\mathsf{ST}_{C^*},b_{\mathrm{com}}) \leftarrow \langle C^*(\rho),R\rangle(1^\lambda)\right] = \mathsf{negl}(\lambda). \tag{7}$$

Assuming Claim 7, it is straightforward to finish the proof of Claim 6 by using Lem. 14. Claim 7 means that the Good occurs whenever $b_{\mathrm{com}}=1$ except for negligible probability. Since $\mathcal{SE}_{\mathsf{Acc}}$ is only required to correctly extract and simulate when $b_{\mathrm{com}}=1$, it suffices to give an extractor that correctly

extracts and simulates when $\{\mathsf{wExtCom.com}_i\}_{i=1}^n$ satisfies the condition for Good. Since $\mathsf{wExtCom}$ satisfies the special parallel weak extractability with ε-simulation as shown in Lem. 14, $\mathcal{SE}_{\mathsf{VSS}_{\mathsf{Recon}}}$ given in Def. 16 (where we set $F := \mathsf{VSS}_{\mathsf{Recon}}$) directly gives $\mathcal{SE}_{\mathsf{Acc}}$. Specifically, $\mathcal{SE}_{\mathsf{Acc}}$ as described below suffices for Claim 6.

$\mathcal{SE}_{\mathsf{Acc}}^{C^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})$:

1. Run $(\{\mathsf{wExtCom.com}_i\}_{i=1}^n, m_{\mathsf{Ext}}, \mathsf{ST}_{C_2^*}) \leftarrow \mathcal{SE}_{\mathsf{VSS}_{\mathsf{Recon}}}^{C_2^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})$ where C_2^* denotes the action of C^* until Step 2 in the commit stage where it outputs $\{\mathsf{wExtCom.com}_i\}_{i=1}^n$.
2. Simulate the interaction between C^* and R from Step 3 where the state of C^* is initialized to be $\mathsf{ST}_{C_2^*}$. Let b_{com} be R's decision (i.e., $b_{\mathrm{com}} = 1$ if and only if R accepts) and ST_{C^*} be the post-execution state of S
3. Output $(m_{\mathsf{Ext}}, \mathsf{ST}_{C^*}, b_{\mathrm{com}})$.

Now, the only thing left is to prove Claim 7.

Proof of Claim 7. This proof follows from a similar argument which has been used to establish the soundness of the [IKOS07] commit-and-prove protocol [IKOS07, GLOV12, GOSV14, LP21].

Let $\mathsf{v}_i^* := \mathsf{val}_{\mathsf{wExtCom}}(\mathsf{wExtCom.com}_i)$ for all $i \in [n]$. We now define an object called *inconsistency graph*. This is an undirected graph G with n vertices, where the i-th vertex corresponds to v_i^*; there is an edge between vertices i and j in G if only only if v_i^* and v_j^* are *inconsistent* (as per Def. 3) w.r.t. the $\mathsf{VSS}_{\mathsf{Share}}$ execution. Let B_G denote that set of vertices that form a *minimum vertex cover*[32] of G (When B_G is not unique, pick one arbitrarily). Next, we prove Eq. (7) by considering the following two possibilities based on the size of B_G:

If $|B_G| \leq t$: we argue that except with negligible probability, either the even Good will happen, or we must have $b_{\mathrm{com}} = 0$.

To see that, consider an execution of $\mathsf{VSS}_{\mathsf{Share}}$ where an adversary corrupts the set of players in B_G, and behaves in a way that the views of any player P_j, for $j \notin B_G$, is v_j^*. Such an execution can be obtained by choosing all the messages from $P_j \in B_G$ to $P_j \notin B_G$ as in the view v_j^*; since B_G is a vertex cover, every pair of views $(\mathsf{v}_i^*, \mathsf{v}_j^*)$ with $i, j \in \overline{B_G}$ are not connected in the graph G and hence consistent. Finally, by the $(n+1, t)$-perfect verifiable-committing property of VSS (see Property 1), such a corruption should not influence the output of the honest players in the $\mathsf{VSS}_{\mathsf{Share}}$ stage. That is, one of the following two cases (corresponding to the two possibilities listed in Property 1) must happen:

1. During the Sharing Phase, all honest players (i.e., those in $\overline{B_G}$) disqualify the dealer. In this case, for each $i \in \overline{B_G}$, v_i^* contains a special symbol $\perp$ indicating the failure of the Sharing Phase. Recall that R checks a size-t random subset (determined by the coin-flipping in Steps 3 to 5) of $\{v_i^*\}_{i\in[n]}$. Since $|\overline{B_G}| > n - t$ and t is a constant fraction of n, R will pick at least one v^*i for $i \in \overline{B_G}$ with overwhelming probability; in this case, R learns the failure of the Sharing Phase and thus rejects the Commit Phase (i.e., $b_{\mathrm{com}} = 0$).
2. During the Sharing Phase, honest players do not disqualify the dealer. Therefore such a phase determines a unique value m^* such that $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1^*, \ldots, \mathsf{v}_n^*) = m^*$, which implies that $\{\mathsf{wExtCom.com}_i\}_{i=1}^n$ is $\mathsf{VSS}_{\mathsf{Recon}}$-good in the sense of Def. 16 (with $\overline{B_G}$ playing the role of the set V in Def. 16). Put in other words, the even Good is happening now.

Remark 6. Notice that in the above argument, it is essential that the verifiable-committing property of VSS is *perfect* (see Def. 1), because it implies that that following types of corruption do not hurt

[32] Recall that a *vertex cover* of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.

the security of VSS: (1) semi-honest corruption; (2) semi-honest corruption with maliciously chosen randomness. Therefore, an "effective corruption" must create inconsistency with at least one honest party. Indeed, if this property is only statistical, extra efforts are needed to finish this proof. See [IKOS07] for details.

If $|B_G| > t$: we argue that $b_{\mathrm{com}} = 1$ (i.e., R accepts at the end of the Commit Stage) with at most negligible probability. Recall that R checks the consistency of a size-t random subset of all the views $\{\mathsf{v}_i^*\}_{i\in[n]}$ (i.e., vertices in G). We only need to argue that such a checking will hit an edge in G with overwhelming probability. For this, we use the well-known connection between the size of a minimum vertex cover to the size of a *maximum matching*. Concretely, the graph G must have a *matching*[33] $\mathcal{M}$ of size at least $t/2$. (Otherwise, if the maximum matching contains less than $t/2$ edges, then the vertices of this matching form a vertex cover set B with $|B| < t$.) Recall that if R hits any edge of G, he will reject. The probability that the t vertices (views) that R picks miss all the edges of G is smaller than the probability that he misses all edges of the matching, which is again at most $2^{-\Omega(t)} = 2^{-\Omega(\lambda)}$. To see that, suppose that the first $t/2$ vertices picked by R do not hit an edge of the matching. Denote this set of vertices as $S_{t/2}$. It follows from Serfling's Inequality (see Lem. 1) that with overwhelming probability over λ, $S_{t/2}$ contains $\Omega(t)$ vertices that are the vertices of the edges $\mathcal{M}$. Then, their $\Omega(t)$ matching neighbors will have $\Omega(t/n) = \Omega(1)$ probability of being hit by each subsequent vertex picked by R. Since R will pick $t/2$ more vertices, the probability that R misses all the $\Omega(t)$ matching neighbors with probability at most $2^{-\Omega(t)} = 2^{-\Omega(\lambda)}$.

This finishes the proof of Claim 7. □

This finishes the proof of Claim 6. □

This eventually concludes the proof of Lem. 16. □

Lemma 17 (Parallel-Strong Extractability with ε-Simulation). ExtCom *is parallel-strongly extractable with ε-simulation (as per Def. 12).*

Proof. Since this lemma can be proven similarly to Lem. 16, we only highlight the differences from that. Similarly to the proof of Lem. 16, by using Watrous' rewinding lemma Lem. 2, we only have to construct an extractor that correctly extracts and simulates when R accepts in all the parallel sessions. That is, it suffices to prove the following lemma.

Claim 8 (Extraction and Simulation for Accepting Case)**.** *For any integer $N = \mathsf{poly}(\lambda)$, there exists a QPT algorithm $\mathcal{SE}_{\mathsf{par},\mathsf{Acc}}$ such that for any noticeable $\varepsilon(\lambda)$ and any non-uniform QPT $C^*(\rho)$,*

$$\left\{(\mathit{\Gamma}_{\{b_{\mathrm{com},j}\}_{j=1}^N}(\{m_{\mathsf{Ext},j}\}_{j=1}^N, \mathsf{ST}_{C^*}) : (\{m_{\mathsf{Ext},j}\}_{j=1}^N, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^N) \leftarrow \mathcal{SE}_{\mathsf{par},\mathsf{Acc}}^{C^*(\rho)}(1^\lambda, 1^{\varepsilon^{-1}})\right\}_\lambda$$
$$\overset{c}{\approx}_\varepsilon \left\{(\mathit{\Gamma}_{\{b_{\mathrm{com},j}\}_{j=1}^N}(\{\mathsf{val}_{\mathsf{ExtCom}}(\mathsf{com}_j)\}_{j=1}^N, \mathsf{ST}_{C^*})) : (\{\mathsf{com}_j\}_{j=1}^N, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^N) \leftarrow \langle C^*(\rho), R^N\rangle(1^\lambda)\right\}_\lambda$$

where $(\{\mathsf{com}_j\}_{j=1}^N, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^N) \leftarrow \langle C^*(\rho), R^N\rangle(1^\lambda)$ means that $C^*(\rho)$ interacts with N copies of the honest receiver R in parallel and the execution results in transcripts $\{\mathsf{com}_j\}_{j=1}^N$, the final state ST_{C^*}, and outputs $\{b_{\mathrm{com},j}\}_{j=1}^N$ of each copy of R and

$$\mathit{\Gamma}_{\{b_{\mathrm{com},j}\}_{j=1}^N}(\{m_j\}_{j=1}^N, \mathsf{ST}_{C^*}) := \begin{cases} (\{m_j\}_{j=1}^N, \mathsf{ST}_{C^*}) & \text{if } \forall\ j \in [N]\ b_{\mathrm{com},j} = 1 \\ \bot & \text{otherwise} \end{cases}.$$

[33] Recall that a matching is a set of edges without common vertices.

Remark 7. One may think that the above claim is similar to the parallel-strong extractability with ε-simulation (Def. 12). However, the crucial difference is that the above claim does not require the extractor to simulate ST_{C^*} when $b_{\mathrm{com},j} = 0$ for some j.

Below, we prove Claim 8 using the special parallel weak extractability with ε-simulation of $\mathsf{wExtCom}$ (Lem. 14) similarly to the proof of Claim 6.

In an N-parallel execution $\langle C^*(\rho), R^N\rangle(1^\lambda)$, Let $\mathsf{wExtCom.com}_{j,i}$ be the i-th commitment of $\mathsf{wExtCom}$ in Step 2 in the commit stage in the j-th session of ExtCom for $i \in [n]$ and $j \in [N]$. In the execution of $(\{\mathsf{com}_j\}_{j=1}^N, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^N) \leftarrow \langle C^*(\rho), R^N\rangle(1^\lambda)$, let Good_j be the event that $\{\mathsf{wExtCom.com}_{j,i}\}_{i=1}^n$ is $\mathsf{VSS}_{\mathsf{Recon}}$-good in the sense of Def. 16, i.e.,

- there exists $V_j \subseteq [n]$ such that $\mathsf{wExtCom.com}_{j,i}$ is valid (i.e., $\mathsf{val}_{\mathsf{wExtCom}}(\mathsf{wExtCom.com}_{j,i}) \neq \bot$) for all $i \in V_j$, and
- there exists $m_j^* \neq \bot$ such that $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}'_1, \dots, \mathsf{v}'_n) = m^*$ for all $(\mathsf{v}'_1, \dots, \mathsf{v}'_n)$ such that

$$\forall i \in V_j,\ \mathsf{v}'_i = \mathsf{val}_{\mathsf{wExtCom}}(\mathsf{wExtCom.com}_{j,i}).$$

Let Bad_j be the complementary event of Good_j. We prove the following claim.

Claim 9. *For all $j \in [N]$, It holds that*

$$\Pr\left[\mathsf{Bad}_j \wedge b_{\mathrm{com},j} = 1 : (\{\mathsf{com}_j\}_{j=1}^N, \mathsf{ST}_{C^*}, \{b_{\mathrm{com},j}\}_{j=1}^N) \leftarrow \langle C^*(\rho), R^N\rangle(1^\lambda)\right] = \mathsf{negl}(\lambda).$$

We can prove Claim 9 in exactly the same way as the proof of Claim 7 by focusing on one session while ignoring all the other sessions.

By Claim 9 and the union bound, Good_j occurs for all $j \in [N]$ simultaneously whenever $b_{\mathrm{com},j} = 1$ for all $j \in [N]$ except for negligible probability. Since $\mathcal{SE}_{\mathsf{par},\mathsf{Acc}}$ is only required to correctly extract and simulate when $b_{\mathrm{com},j} = 1$ for all $j \in [N]$, it suffices to give an extractor that correctly extracts and simulates when Good_j occurs, i.e., $\{\mathsf{wExtCom.com}_{j,i}\}_{i=1}^n$ is $\mathsf{VSS}_{\mathsf{Recon}}$-good for all $j \in [N]$. In this case, it is easy to see that $(\{\mathsf{wExtCom.com}_{1,i}\}_{i=1}^n, ..., \{\mathsf{wExtCom.com}_{N,i}\}_{i=1}^n)$ is F-good if we define

$$F((\mathsf{v}_{1,1}, ..., \mathsf{v}_{1,n}), ..., (\mathsf{v}_{N,1}, ..., \mathsf{v}_{N,n})) := (\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_{1,1}, ..., \mathsf{v}_{1,n}), ..., \mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_{N,1}, ..., \mathsf{v}_{N,n}))$$

with the corresponding subset

$$V := V_1 \times V_2 ... \times V_N.$$

Thus, we can prove Claim 8 by using the special parallel weak extractability with ε-simulation of $\mathsf{wExtCom}$ (Lem. 14) similarly to the proof of Claim 6. □

6 Black-Box ε-Simulatable ExtCom-and-Prove in Constant Rounds

6.1 Definition

The following definition is taken from [CLP20, LP21] with modifications to admit an ε-simulation-extractable Commit Stage and an ε-ZK Prove Stage.

Definition 17 (ε-Simulatable ExtCom-and-Prove). *An ε-Simulatable ExtCom-and-Prove scheme consists of a pair of protocols $\Pi_{\mathrm{ECNP}} = (\mathsf{ExtCom}, \mathsf{Prove})$ executed between a pair of* PPT *machines P and V. Let $m \in \{0,1\}^{\ell(\lambda)}$ (where $\ell(\cdot)$ is some polynomial) is a message that P wants to commit to. The protocol consists of the following stages (we omit the input 1^λ to P and V):*

– **Commit Stage:** *$P(m)$ and V execute* ExtCom*, which generates a transcript (commitment)* com*, P's state* ST_P*, and V's decision bit $b \in \{0,1\}$ indicating acceptance (i.e., $b = 1$) or rejection (i.e., $b = 0$). We denote this execution as* $(\mathsf{com}, \mathsf{ST}_P, b) \leftarrow \langle P(m), V\rangle_{\mathsf{EC}}$*. A malicious verifier is allowed to output any quantum state, which we denote by* ST_{V^*} *instead of b, and to keep the state for the prove stage.*

– **Decommit Stage:**[34] *$P(\mathsf{ST}_P)$ generates a decommitment* decom *and sends it to V along with a message m. V accepts or rejects.*

– **Prove Stage:** *Let ϕ be any predicate. $P(\mathsf{ST}_P, \phi)$ and $V(\mathsf{com}, \phi)$ execute* Prove*, after which V outputs 1 (accept) or 0 (reject). We denote the execution of this stage as $b' \leftarrow \langle P(\mathsf{ST}_P), V(\mathsf{com})\rangle^{\phi}_{\mathsf{Pr}}$, where $b' \in \{0,1\}$ is V's output. A malicious verifier is allowed to output an arbitrary quantum state, which we denote by* OUT_{V^*} *instead of b'.*

The following requirements are satisfied:

1. **Security as ε-Simulation Extractable Commitment.** *The Commit Stage and Decommit Stage constitute a post-quantum commitment scheme (as per Def. 7 where P and V play the roles of C and R, respectively) that is computationally hiding (as per Def. 8), statistically binding (as per Def. 9), and strongly extractable with ε-simulation (as per Def. 11).*

2. **Completeness.** *For any $m \in \{0,1\}^{\ell(\lambda)}$ and any polynomial-time computable predicate ϕ s.t. $\phi(m) = 1$, it holds that*
$$\Pr\left[b = 1 \ \wedge \ b' = 1 : \begin{array}{l}(\mathsf{com}, \mathsf{ST}_P, b) \leftarrow \langle P(m), V\rangle_{\mathsf{EC}} \\ b' \leftarrow \langle P(\mathsf{ST}_P), V(\mathsf{com})\rangle^{\phi}_{\mathsf{Pr}}\end{array}\right] = 1. \tag{8}$$

3. **Soundness.** *For any predicate ϕ and any non-uniform QPT prover $P^*(\rho)$,*
$$\Pr\left[\begin{array}{l}b = 1 \ \wedge \ b' = 1 \\ \wedge \ \phi(\mathsf{val}_{\mathsf{ExtCom}}(\mathsf{com})) = 0\end{array} : \begin{array}{l}(\mathsf{com}, \mathsf{ST}_{P^*}, b) \leftarrow \langle P^*(\rho), V\rangle_{\mathsf{EC}} \\ b' \leftarrow \langle P^*(\mathsf{ST}_{P^*}), V(\mathsf{com})\rangle^{\phi}_{\mathsf{Pr}}\end{array}\right] = \mathsf{negl}(\lambda), \tag{9}$$
where $\mathsf{val}_{\mathsf{ExtCom}}(\mathsf{com})$ is as defined in Def. 10 and we stipulate that $\phi(\bot) = 0$.

4. **ε-Zero-Knowledge.** *There exists a pair of QPT simulators $(\mathcal{S}_{\mathsf{EC}}, \mathcal{S}_{\mathsf{Pr}})$ such that for any $m \in \{0,1\}^{\ell(\lambda)}$, polynomial-time computable predicate ϕ s.t. $\phi(m) = 1$, any non-uniform QPT verifier $V^*(\rho)$, and any noticeable function $\varepsilon(\lambda)$, the following conditions hold:*
$$\left\{\widetilde{\mathsf{ST}}_{V^*} : (\widetilde{\mathsf{ST}}_{V^*}, \mathsf{ST}_{\mathsf{EC}}) \leftarrow \mathcal{S}^{V^*(\rho)}_{\mathsf{EC}}\right\}_{\lambda} \overset{c}{\approx} \left\{\mathsf{ST}_{V^*} : (\mathsf{com}, \mathsf{ST}_P, \mathsf{ST}_{V^*}) \leftarrow \langle P(m), V^*(\rho)\rangle_{\mathsf{EC}}\right\}_{\lambda} \tag{10}$$
$$\left\{\begin{array}{l}\widetilde{\mathsf{OUT}}_{V^*} : \begin{array}{l}(\widetilde{\mathsf{ST}}_{V^*}, \mathsf{ST}_{\mathsf{EC}}) \leftarrow \mathcal{S}^{V^*(\rho)}_{\mathsf{EC}} \\ \widetilde{\mathsf{OUT}}_{V^*} \leftarrow \mathcal{S}^{V^*}_{\mathsf{Pr}}(1^{\varepsilon^{-1}}, \widetilde{\mathsf{ST}}_{V^*}, \mathsf{ST}_{\mathsf{EC}}, \phi)\end{array}\end{array}\right\}_{\lambda} \overset{c}{\approx}_{\varepsilon} \left\{\mathsf{OUT}_{V^*} : \begin{array}{l}(\mathsf{com}, \mathsf{ST}_P, \mathsf{ST}_{V^*}) \leftarrow \langle P(m), V^*(\rho)\rangle_{\mathsf{EC}} \\ \mathsf{OUT}_{V^*} \leftarrow \langle P(\mathsf{ST}_P), V^*(\mathsf{ST}_{V^*})\rangle^{\phi}_{\mathsf{Pr}}\end{array}\right\}_{\lambda}. \tag{11}$$
We refer to $\mathcal{S}_{\mathsf{EC}}$ (resp. $\mathcal{S}_{\mathsf{Pr}}$) as the Commit-Stage (resp. Prove-Stage) simulator.

Remark 8 (On the ZK Conditions). Eq. (10) is optional. We include it because our construction achieves it. The ε-zero-knowledge property defined by Eq. (11) alone should suffice for most applications.

In Lem. 18, we show the existence of a construction satisfying Def. 17, only assuming black-box access to post-quantum secure OWFs.

Lemma 18. *Assume the existence of post-quantum secure OWFs. Then, there exists a constant-round construction of Π_{ECNP} satisfying Def. 17. Moreover, this construction makes only black-box use of the assumed OWF.*

We will prove Lem. 18 by presenting the construction (and security proof) in Sec. 6.5. Before that, we will first present three applications of such an ε-simulatable ExtCom-and-Prove protocol in Sec. 6.2 to 6.4.

[34] This stage is rarely executed in applications.

6.2 Application I: ε-Simulatable Coin-Flipping Protocols

Let us first recall a canonical construction of two-party coin-flipping protocol from any extractable-and-equivocal string-commitment $\mathsf{EqExtCom}$ in the classical setting ([CF01, PW09]). To toss $\ell(\lambda)$ coins, this construction works in 3 steps: (1) P_1 commits to a random string $r_1 \in \{0,1\}^{\ell(\lambda)}$ using $\mathsf{EqExtCom}$; (2) P_2 sends to P_1 a random string $r_2 \in \{0,1\}^{\ell(\lambda)}$; (3) P_1 then decommits to the r_1 he committed to in step (1). Both parties set $r := r_1 \oplus r_2$ as the coin-flipping result.

This protocol flips $\ell(\lambda)$ coins securely as per the simulation-based definition: To simulate for a malicious P_1^*, the simulator $\mathcal{S}$ extracts from step (1) the value r_1^* committed by P_1^* relying on the extractability of $\mathsf{EqExtCom}$. $\mathcal{S}$ then sends $r_2 = r_1^* \oplus \widetilde{r}$ as the simulated step-(2) message, where $\widetilde{r}$ is the random string $\mathcal{S}$ received from the ideal functionality in the ideal world (i.e., $\mathcal{S}$ needs to "bias" the coin-tossing result to $\widetilde{r}$). To simulate for a malicious P_2^*, $\mathcal{S}$ first commit to an arbitrary string of length $\ell(\lambda)$ using $\mathsf{EqExtCom}$, as the simulated step-(1) message; then in step (3), relying on the equivocality of $\mathsf{EqExtCom}$, $\mathcal{S}$ "decommits" the step-(1) value to $r_2^* \oplus \widetilde{r}$, where r_2^* is the step-(3) message received from P_2^* and $\widetilde{r}$ again is the random string $\mathcal{S}$ obtained from the ideal functionality.

Our post-quantum ε-simulatable coin-flipping protocol follows the above classical protocol but with the following modification. We observe that the ε-simulatable ExtCom-and-Prove protocol Π_{ECNP} can be used to achieve the same effect as a extractable-and-equivocal commitment (albeit with ε-simulation). To see that, we just need to "interpret" Π_{ECNP} in the following way:

- **Commit:** $C(m)$ and R simply run the Commit Stage of Π_{ECNP}, where C commits to its message m.
- **Decommit:** To decommit to a message m, C sends to R the message m only (*without decommitment*); then, they execute the Prove Stage of Π_{ECNP} where C proves a special predicate the $\phi_m(\cdot)$, which equals to 1 if and only if the input equals to (the hard-wired) m.

Such a commitment is extractable with ε-simulation as the Commit Stage of Π_{ECNP} is so. Also, it is equivocal with ε-simulation because Π_{ECNP} is ε-zero-knowledge. That is, a equivocator $\mathcal{E}$ can be constructed by running the ε-ZK simulator guaranteed by Property 4. We present in Prot. 3 our coin-flipping protocol constructed in black-box from Π_{ECNP}.

Protocol 3: ε-Simultable Coin-Flipping

Let $\ell(\lambda)$ be a polynomial specifying the length of the desired coin-flipping result. Both parties only take the security parameter 1^λ as their input.

1. P_1 and P_2 execute the Commit Stage of Π_{ECNP}, where P_1 commits to a random string $r_1 \in \{0,1\}^{\ell(\lambda)}$;
2. P_2 samples a random string $r_2 \leftarrow \{0,1\}^{\ell(\lambda)}$ and sends it to P_1;
3. P_1 sends r_1 to P_2, and then uses the Prove Stage of Π_{ECNP} to prove to P_2 the predicate $\phi_{r_1}(\cdot)$, which equals 1 if and only if the input equals r_1.

Both parties set $r := r_1 \oplus r_2$ as the coin-flipping result.

On Security. The security of Prot. 3 can be proved following a similar argument as the above one in the classical setting: If P_1^* is corrupted, we build the simulator $\mathcal{S}$ by extracting r_1^* from Step 1 using the ε-simulation extractor for the Commit Stage of Π_{ECNP} (of course, setting $r_2 = r_1^* \oplus \widetilde{r}$ as in the above classical setting). Since such an extractor ensures that the state of P_1^* after extraction is at most ε-far from the real execution, $\mathcal{S}$ works as expected. If P_2^* is corrupted, $\mathcal{S}$ will simulate the Step 1 by committing to an arbitrary string of length $\ell(\lambda)$, and then "equivocate" r_1 to $r_2 \oplus \widetilde{r}$ in Step 3 using the simulator $\mathcal{S}_{\text{ECNP}}$ of Π_{ECNP} (i.e., it simulates a proof for the correctness of ϕ_{r_1}).

It then follows from the ε-ZK property of Π_{ECNP} (more accurately, Eq. (11)) that $\mathcal{S}$ is a proper ε-simulator for the corrupted P_2^*. We suppress further details as this argument is standard. In summary, we have the following corollary of Lem. 18.

Corollary 1. *Assume the existence of post-quantum secure OWFs. Then, there exists a constant-round construction of ε-simulatable coin-flipping. Moreover, this construction makes only black-box use of the assumed OWF.*

6.3 Application II: ZKAoK with ε-Simulatable Knowledge Extractor

Definitions. We define post-quantum argument for **NP** and its quantum ε-zero-knowledge property and argument of knowledge with ε-simulation extractor property. For a language $\mathcal{L} \in \mathbf{NP}$, let $\mathcal{R}_{\mathcal{L}}$ be the corresponding relation function, i.e., $\mathcal{R}_{\mathcal{L}}(x, w) = 1$ if and only if w is a valid witness of x. We write $\mathcal{R}_{\mathcal{L}}(x)$ to mean the set of all valid witnesses for the statement x. .

Definition 18 (Post-Quantum Argument Systems for NP). *A classical protocol (P, V) with an honest PPT prover P and an honest PPT verifier V for a language $\mathcal{L} \in \mathbf{NP}$ is said to be* post-quantum argument *if it satisfies the following requirements where OUT_V denotes the final output of V.*

1. **Completeness.** *For any $x \in \mathcal{L}$ and any $w \in \mathcal{R}_{\mathcal{L}}(x)$,*
$$\Pr[\mathsf{OUT}_V\langle P(w), V\rangle(x) = 1] \geq 1 - \mathsf{negl}(\lambda).$$
2. **Computational Soundness:** *For any quantum polynomial-size prover $P^* = \{P_\lambda^*, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ and any $x \in \{0,1\}^\lambda \setminus \mathcal{L}$,*
$$\Pr[\mathsf{OUT}_V\langle P_\lambda^*(\rho_\lambda), V\rangle(x) = 1] \leq \mathsf{negl}(\lambda).$$

Definition 19 (Quantum ε-ZK for NP). *Let (P, V) be a post-quantum argument for a language $\mathcal{L} \in \mathbf{NP}$ as in Def. 18. The protocol is quantum ε-zero-knowledge if it satisfies:*

1. **Quantum ε-Zero-Knowledge.** *There exists an oracle-aided QPT simulator $\mathcal{S}$, such that for any quantum polynomial-size verifier $V^* = \{V_\lambda^*, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ and any noticeable function $\varepsilon(\lambda)$,*
$$\left\{\mathsf{OUT}_{V_\lambda^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)\right\}_{\lambda,x,w} \overset{c}{\approx}_\varepsilon \left\{\mathcal{S}^{V_\lambda^*(\rho_\lambda)}(1^{\varepsilon^{-1}}, x)\right\}_{\lambda,x,w},$$
where $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_{\mathcal{L}}(x)$, and $\mathsf{OUT}_{V_\lambda^*}$ denotes V_λ^*'s final output.

Definition 20 (Argument of Knowledge with ε-Simulation Extractor). *A post-quantum argument system (P, V) for an **NP** language $\mathcal{L}$ is an argument of knowledge with ε-simulation extractor if there exists a QPT machine $\mathcal{SE}$ such that the following holds: for any non-uniform QPT machine $P^*(\rho)$, any $x \in \{0,1\}^\lambda$, and any noticeable function $\varepsilon(\cdot)$, the following condition is satisfied:*
$$\left\{(\widetilde{\mathsf{ST}}_{P^*}, \mathcal{R}_{\mathcal{L}}(x, \widetilde{w})) : (\widetilde{\mathsf{ST}}_{P^*}, \widetilde{w}) \leftarrow \mathcal{SE}^{P^*}(1^{\varepsilon^{-1}}, x, \rho)\right\}_\lambda \overset{c}{\approx}_\varepsilon \left\{(\mathsf{ST}_{P^*}, b) : (\mathsf{ST}_{P^*}, b) \leftarrow \langle P^*(x, \rho), V(x)\rangle\right\}_\lambda.$$

Our Construction. To give an ε-ZK protocol also satisfying Def. 20, we first recall a canonical (constant-round but non-black-box) construction in the classical setting: the prover commits to the witness w using an extractable commitment, and then proves using a zero-knowledge protocol to the verifier that the committed value is a valid witness for the concerned statement x.

Our construction is obtained by observing that in the above protocol, what the prove does is essentially an ExtCom-and-Prove. I.e., the prover's initial commitment can be interpreted as the Commit Stage of the ExtCom-and-Prove, committing to the witness w. The subsequent zero-knowledge protocol can be viewed as executing the Prove Stage of the ExtCom-and-Prove where the prover prove a special predicate $\phi_x(\cdot)$, for which $\phi_x(w)$ if and only if its input w is a valid witness w (i.e., $\mathcal{R}_{\mathcal{L}}(x, w) = 1$). We present the construction in Prot. 4.

Protocol 4: Zero-Knowledge Argument of Knowledge with ε-Simulation Extractor

Input: both parties have the statement x and the security parameter 1^λ as the common input; P additionally obtains the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ as its private input.

1. P and V execute the Commit Stage of Π_{ECNP}, where P commits to w.
2. P and V execute the Prove Stage of Π_{ECNP}, where P proves the special predicate $\phi_x(\cdot)$, for which $\phi_x(w) = 1$ if and only if $\mathcal{R}_{\mathcal{L}}(x, w) = 1$.

Verifier's Decision. V accepts if and only if both the Commit and Prove Stage are convincing.

On Security. It is easy to see that soundness of Prot. 4 follows from the soundness of Π_{ECNP} (Property 3), and that ε-ZK property of Prot. 4 follows from that of Π_{ECNP} (in particular, Eq. (11) in Property 4). To show that Prot. 4 is a argument of knowledge with ε-simulation extractor (as per Def. 20), we rely on the ε-simulatable extractability (Property 1) of the Commit Stage of Π_{ECNP}. In more detail, the ε-simulation knowledge extractor $\mathcal{SE}$ can be construct as follows: $\mathcal{SE}$ extracts the value committed by the malicious prover P^* in Step 1, while also performing a ε-close simulation for P^*'s internal state after the extraction. We emphasize that the Commit Stage of Π_{ECNP} is a ε-simulatable *strongly* extractable commitment (as per Def. 11); that is, $\mathcal{SE}$ (using the ε-simulation extractor ensured by Def. 11) can always extract the value committed in Step 1 while performing a ε-simulation for P^*'s state, *even if the Step 1 commitment is invalid (in which case, the committed value is define as $\perp$).* Thus, Def. 20 will be satisfied. Since the above arguments for security are standard, we omit the details. In summary, we obtain the following corollary of Lem. 18.

Corollary 2. *Assume the existence of post-quantum secure OWFs. Then, there exists a constant-round construction of ε-zero-knowledge argument of knowledge with an ε-simulation knowledge extractor for* **NP**. *Moreover, this construction makes only black-box use of the assumed OWF.*

6.4 Application III: Black-Box ε-ZK for QMA

Definitions. We first present the definition of **QMA**. Note that we formalize **QMA** problems as promise problems. This is because the most ZK-friendly **QMA**-complete problem (known currently) is the Consistency of Local Density Matrices (CLDM) problem, which is in the form of a promise problem. We refer interested readers to [BG20] for details.

Definition 21 (QMA). *We say that a promise problem $\mathcal{L} = (\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}})$ is in* **QMA** *if there is a polynomial ℓ and a QPT algorithm V such that the following is satisfied:*

- *For any $x \in \mathcal{L}_{\text{yes}}$, there exists a quantum state w of $\ell(|x|)$-qubit (called a witness) such that we have $\Pr[V(x, w) = 1] \geq 2/3$.*
- *For any $x \in \mathcal{L}_{\text{no}}$ and any quantum state w of $\ell(|x|)$-qubit, we have $\Pr[V(x, w) = 1] \leq 1/3$.*

For any $x \in \mathcal{L}_{\text{yes}}$, we denote by $R_{\mathcal{L}}(x)$ to mean the (possibly infinite) set of all quantum states w such that $\Pr[V(x, w) = 1] \geq 2/3$.

Next, we define quantum ε-ZK for **QMA**. This definition is taken from [BS20] with modifications to accommodate the ε-simulation.

Definition 22 (Quantum Proof and Argument Systems for QMA). *A quantum protocol (P, V) with an honest QPT prover P and an honest QPT verifier V for a promise problem $\mathcal{L} = (\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}}) \in$* **QMA** *is said to be a quantum proof or argument system if it satisfies the following requirements where OUT_V denotes V's final output.*

1. **Completeness.** *There is a polynomial $k(\cdot)$ s.t. for any $x \in \mathcal{L}_{\text{yes}}$ and any $w \in \mathcal{R}_{\mathcal{L}}(x)$,*

$$\Pr\left[\mathsf{OUT}_V\langle P(w^{\otimes k(\lambda)}), V\rangle(x) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

2. **Soundness:** *the protocol satisfies one of the following*

   – **Computational Soundness:** *For any quantum polynomial-size prover $P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ and any $x \in \mathcal{L}_{\text{no}} \cap \{0,1\}^\lambda$,*

   $$\Pr[\mathsf{OUT}_V\langle P^*_\lambda(\rho_\lambda), V\rangle(x) = 1] \leq \mathsf{negl}(\lambda).$$

   A protocol with computational soundness is called an argument.

   – **Statistical Soundness:** *For any (potentially unbounded) P^* and any $x \in \mathcal{L}_{\text{no}} \cap \{0,1\}^\lambda$,*

   $$\Pr[\mathsf{OUT}_V\langle P^*, V\rangle(x) = 1] \leq \mathsf{negl}(\lambda).$$

   A protocol with statistical soundness is called a proof.

Definition 23 (Quantum ε-ZK for QMA). *Let (P,V) be a quantum protocol (argument or proof) for a language $\mathcal{L} \in$ **QMA** as in Def. 22, where the prover uses $k(\lambda)$ copies of a witness. The protocol is quantum ε-zero-knowledge if it satisfies:*

1. **Quantum ε-Zero-Knowledge.** *There exists an oracle-aided QPT simulator $\mathcal{S}$, such that for any quantum polynomial-size verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ and any noticeable function $\varepsilon(\lambda)$,*

$$\left\{\mathsf{OUT}_{V^*_\lambda}\langle P(w^{\otimes k(\lambda)}), V^*_\lambda(\rho_\lambda)\rangle(x)\right\}_{\lambda,x,w} \overset{c}{\approx}_\varepsilon \left\{\mathcal{S}^{V^*_\lambda(\rho_\lambda)}(1^{\varepsilon^{-1}}, x)\right\}_{\lambda,x,w},$$

   where $\lambda \in \mathbb{N}$, $x \in \mathcal{L}_{\text{yes}} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_{\mathcal{L}}(x)$, and $\mathsf{OUT}_{V^*_\lambda}$ denotes the V^*_λ's final output.

We present in Def. 24 the definition of quantum sigma protocols for **QMA**, which will be used as a building block for our construction. This definition is again take from [BS20]. We emphasize that the verifier's message β must be a classical string. As observed in [BS20], the parallel version of [BG20] satisfies Def. 24.

Definition 24 (Quantum Sigma Protocol for QMA). *A quantum sigma protocol for $\mathcal{L} = (\mathcal{L}_{\text{yes}}, \mathcal{L}_{\text{no}}) \in$ **QMA** is a quantum proof system $(\Xi.P, \Xi.V)$ (as per Def. 22) with 3 messages and the following syntax.*

– $(\alpha, \tau) \leftarrow \Xi.P_1(x, w^{\otimes k(\lambda)})$: *Given an $x \in \mathcal{L}_{\text{yes}} \cap \{0,1\}^\lambda$ and $k(\lambda)$ witnesses $w \in \mathcal{R}_{\mathcal{L}}(x)$ (for a polynomial $k(\cdot)$), the first prover execution outputs a public message α for $\Xi.V$ and a private inner state τ.*

– $\beta \leftarrow \Xi.V(x)$: *The verifier simply outputs a* classical string *of $\mathsf{poly}(|x|)$ random bits.*

– $\gamma \leftarrow \Xi.P_3(\beta, \tau)$: *Given the verifier's string β and the private state τ, the prover outputs a response γ.*

The protocol satisfies the following:

1. **Special Zero-Knowledge:** *There exists a QPT simulator $\Xi.\mathcal{S}$ such that,*

$$\left\{(\alpha,\gamma) : (\alpha,\tau) \leftarrow \Xi.P_1(x, w^{\otimes k(\lambda)}); \gamma \leftarrow \Xi.P_3(\beta,\tau)\right\}_{\lambda,x,w,\beta} \overset{c}{\approx} \left\{(\alpha,\gamma) : (\alpha,\gamma) \leftarrow \Xi.\mathcal{S}(x,\beta)\right\}_{\lambda,x,w,\beta},$$

   where $\lambda \in \mathbb{N}$, $x \in \mathcal{L}_{\text{yes}} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_{\mathcal{L}}(x)$, and $\beta \in \{0,1\}^{\mathsf{poly}(\lambda)}$.

The [BS20] Protocol. Since our construction is essentially a black-box version of the protocol from [BS20, Section 5], it is helpful to recall the [BS20] protocol and its security proof. We will show how to make it black-box (but ε-ZK) using our black-box ε-simulatable ExtCom-and-Prove.

In the [BS20] construction, the verifier first executes the $\beta \leftarrow \Xi.V(x)$ algorithm of the quantum sigma protocol defined in Def. 24, and commits to β using a post-quantum simulation-extractable commitment. Then, the prover and the verifier will execute the quantum sigma protocol but with one modification: in the 2nd round of the quantum sigma protocol (where the verifier is supposed to send his message β), the verifier will send the value β that he committed at the beginning *without decommitment*, and then give a post-quantum ZK argument to convince the prover that the β he sends is indeed the one he committed earlier; Other parts of the quantum sigma protocol are executed as they should be.

To see why this construction is sound, consider a hybrid where the verifier commits at the beginning an arbitrary value (say, an all-0 string) of proper length. Then, in the 2nd round of the quantum sigma protocol, the verifier generates β and sends it to the (malicious) prover; the verifier uses the zero-knowledge simulator to "fake" the ZK argument for the consistency between β and his initial commitment. Because of the hiding property of the verifier's commitment and the zero-knowledge property of the ZK argument, this hybrid is indistinguishable with the real execution from the prover's point of view. However, notice in this hybrid that we successfully delayed the sampling of β to the 2nd round of the quantum sigma protocol. Thus, the soundness of this protocol can be reduced to that of the quantum sigma protocol in a straightforward manner.

To show that the construction is zero-knowledge, a simulator will extract the β from the (malicious) verifier's initial commitment; this can be done without being noticed by the malicious verifier because of the simulatable-extractability of the verifier's commitment. That is, the simulator learns β even before the quantum sigma protocol started; moreover, such an extraction only disturbs the state of the malicious verifier negligibly. Therefore, the ZK property can be reduced to the special-ZK property (Property 1) of the quantum sigma protocol using a standard argument.

Our Construction. The [BS20] protocol is not black-box because: (1) the verifier runs a zero-knowledge argument on a *cryptographic* statement (i.e., β is the value committed in the beginning); (2) the constant-round post-quantum zero-knowledge protocol constructed in [BS20] is not black-box; indeed, they use non-black-box techniques both for the construction and simulation.

Our protocol is obtained by observing that what the verifier does in the [BS20] protocol is exactly a fully-simulatable ExtCom-and-Prove. I.e., the verifier's initial commitment can be viewed as a simulation-extractable commitment to β, constituting the Commit Stage; later when he sends β and proofs the consistency, he is essentially giving a post-quantum ZK proving the value in his initial commitment satisfies a special predicate $\phi_\beta(\cdot)$, for which $\phi_\beta(x) = 1$ if and only if $x = \beta$.

Thus, we can simply replace the verifier's commitment and zero-knowledge argument with our black-box ε-simulatable ExtCom-and-Prove protocol Π_{ECNP}. Notice that the Prove Stage of Π_{ECNP} is black-box on the commitment of its Commit Stage; also, the construction of the Prove Stage itself is black-box (albeit ε-ZK, instead of fully-simulatable ZK). This bypasses the aforementioned two sources of non-black-boxness in the [BS20] protocol.

It is also worth noting that our construction does not change the structure of the [BS20] protocol. We simply replace the non-black-box components with a black-box counterpart. Therefore, the security proof of the [BS20] protocol extends smoothly to our construction (with slight modifications to accommodate the ε-simulation in proving the ZK property). Therefore, we suppress further details for the security proof of our construction. For completeness, we present our construction in Prot. 5, which makes black-box use of a quantum sigma protocol $(\Xi.P, \Xi.V)$ as per Def. 24, and our ε-simulatable ExtCom-and-Prove Π_{ECNP} as per Def. 17.

Protocol 5: Quantum ε-Zero-Knowledge Argument for QMA

Common Input: An instance $x \in \mathcal{L}_{\text{yes}} \cap \{0,1\}^{\lambda}$, for security parameter $\lambda \in \mathbb{N}$.

Prover's private input: Polynomially many identical witnesses for x: $w^{\otimes k(\lambda)}$ s.t. $w \in \mathcal{R}_{\mathcal{L}}(x)$;

1. V computes $\beta \leftarrow \Xi.V(x)$. V and P then execute the Commit Stage of Π_{ECNP}, where V commits to β;
2. P computes $(\alpha, \tau) \leftarrow \Xi.P_1(x, w^{\otimes k(\lambda)})$ and sends α to V;
3. V sends β to P. Now, both parties agree on a predicate $\phi_\beta(\cdot)$, for which $\phi_\beta(x) = 1$ if and only if $x = \beta$;
4. V and P execute the Prove Stage of Π_{ECNP}, where V proves the predicate ϕ_β defined in the last step. If the argument was not convincing; P terminates communication outputting $\perp$; otherwise, the protocol continues.
5. P computes $\gamma \leftarrow \Xi.P_3(\beta, \tau)$ and sends γ;

Verifier's Decision: V accepts if and only if $1 = \Xi.V(\alpha, \beta, \gamma)$.

6.5 Our Construction of ExtCom-and-Prove (Proof of Lem. 18)

The construction is shown in Prot. 6. It makes black-box use of the following building blocks:

1. The ε-simulatable, *parallel-strong* extractable (as per Def. 12) commitment ExtCom constructed in Sec. 5.2, which in turn makes black-box use of any post-quantum secure OWFs.
2. A statistically-binding, computationally-hiding (against QPT adversaries) commitment Com. This is also known assuming only black-box access to post-quantum secure OWFs.
3. A $(n+1, t)$-perfectly secure verifiable secret sharing scheme $\mathsf{VSS} = (\mathsf{VSS}_{\mathsf{Share}}, \mathsf{VSS}_{\mathsf{Recon}})$ (see Sec. 3.3);
4. A (n, t)-perfectly secure MPC protocol Π_{MPC} (see Sec. 3.4);

For the VSS and MPC protocols, we require that t is a constant fraction of n such that $t \leq n/3$. There are information-theoretical constructions satisfying these properties [BGW88, CDD+99].

Protocol 6: ε-Simulatable ExtCom-and-Prove

Parameter Setting: Let $n(\lambda)$ be a polynomial on λ. Let t be a constant fraction of n such that $t \leq n/3$.

Input: Both P and the receiver V get 1^{λ} as the common input; P gets a string $m \in \{0,1\}^{\ell(\lambda)}$ as his private input, where $\ell(\cdot)$ is a polynomial.

Commit Stage:

1. P emulates $n+1$ (virtual) players $\{P_i\}_{i\in[n+1]}$ to execute the $\mathsf{VSS}_{\mathsf{Share}}$ protocol "in his head", where the input to P_{n+1} (i.e., the Dealer) is m. Let $\{\mathsf{v}_i\}_{i\in[n+1]}$ be the views of the $n+1$ players describing the execution.
2. P and V involve in n executions of ExtCom in parallel, where in the i-th instance ($i \in [n]$), P commits to v_i.

Decommit Stage:

1. P sends $\{\mathsf{v}_i\}_{i\in[n]}$ together with the corresponding decommitment information w.r.t. the ExtCom in Step 2 of the Commit Stage.

2. V checks that all the decommitments in Step 1 of the Decommit Stage are valid. If so, V outputs $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1, \ldots, \mathsf{v}_n)$ and then halts; otherwise, V outputs $\perp$ and then halts.

Prove Stage: both parties learn a polynomial-time computable predicate ϕ.

1. P emulates "in his head" n (virtual) players $\{P_i\}_{i\in[n]}$, where P_i's input is v_i (from Step 1 of the Commit Stage). These n parties execute Π_{MPC} for the following functionality: the functionality reconstructs $m' := \mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1, \ldots, \mathsf{v}_n)$ and sends the value $\phi(m')$ to all the parties as their output. For $i \in [n]$, let v'_i be the view of party P_i during Π_{MPC}.
2. P and V involve in n executions of Com in parallel, where in the i-th instance ($i \in [n]$), P commits to v'_i.
3. V picks a random string r_1 and commits to it using ExtCom.
4. P picks a random string r_2 and sends it to V.
5. V sends to P the value r_1 together with the corresponding decommitment information w.r.t. the ExtCom in Step 3. Now, both parties learn a coin-tossing result $r = r_1 \oplus r_2$, which specifies a size-t random subset $T \subseteq [n]$.
6. P sends to V in *one round* the following messages:
   (a) $\{\mathsf{v}_i\}_{i\in T}$ together with the corresponding decommitment information w.r.t. the ExtCom in Step 2 of the Commit Stage; **and**
   (b) $\{\mathsf{v}'_i\}_{i\in T}$ together with the corresponding decommitment information w.r.t. the Com in Step 2 of the Prove Stage.
7. V checks the following conditions:
   (a) All the decommitments in Steps 6a and 6b are valid; **and**
   (b) for any $i \in T$, v_i is the prefix of v'_i ; **and**
   (c) for any $i, j \in T$, views $(\mathsf{v}'_i, \mathsf{v}'_j)$ are consistent (as per Def. 3 and Rmk. 2) w.r.t. the $\mathsf{VSS}_{\mathsf{Share}}$ execution in Step 1 of the Commit Stage and the Π_{MPC} execution as described in Step 1 of the Prove Stage.

   If all the checks pass, V accepts; otherwise, V rejects.

Proof of Security. We now prove that Prot. 6 satisfies Def. 17. It is straightforward to see that Prot. 6 is constant-round and makes only black-box access to OWFs. Completeness follows from that of VSS, ExtCom, Com, and Π_{MPC}. In the following, we show ε-simulatable extractability (in Lem. 19), soundness (in Lem. 20), and ε-zero-knowledge (in Lem. 21).

Lemma 19 (ε-Simulation Extractability). *Assume* ExtCom *is parallel-strongly extractable with ε-simulation (as per Def. 12). Then, Prot. 6 satisfies security as ε-simulation extractable commitment defined in Property 1 in Def. 17.*

Proof. First, notice that the statistically-binding property and computationally-hiding property follows straightforwardly from those of ExtCom via standard argument. In the following, we prove strong extractability with ε-simulation.

A commitment com generated in the commit stage of Prot. 6 consists of n commitments of ExtCom, which we denote by $\{\mathsf{ExtCom.com}_i\}_{i=1}^n$. By the statistical binding property of ExtCom, $\mathsf{ExtCom.com}_i$ can be opened to only $\mathsf{v}_i := \mathsf{val}_{\mathsf{ExtCom}}(\mathsf{ExtCom.com}_i)$ for all $i \in [n]$ except for negligible probability. Then, by the definition of the decommit stage of Prot. 6, we can see that $\mathsf{val}_{\text{Prot. 6}}(\mathsf{com}) = \mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1, ..., \mathsf{v}_n)$ where we define $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1, ..., \mathsf{v}_n)$ to be $\perp$ when one of v_i's is $\perp$. Moreover, remark that the verifier (which plays the role of a receiver as an extractable commitment scheme) of Prot. 6 accepts in the commit stage if and only if the prover passes verification of the commit stage of ExtCom in all the sessions and thus $\mathsf{val}_{\text{Prot. 6}}(\mathsf{com}) = \perp$ when the verifier rejects in any of the sessions. Thus, the extractor

for the parallel strong extractability with ε-simulation of ExtCom can be directly used as an extractor for the strong extractability with ε-simulation of Prot. 6. □

Lemma 20 (Soundness). *Assume* ExtCom *and* Com *are statistically binding,* ExtCom *is computationally-hiding,* VSS *is* $(n+1,t)$*-perfectly verifiable-committing (see Def. 1) and* Π_{MPC} *is* (n,t)*-perfectly robust (see Def. 6). Then, Prot. 6 satisfies the soundness defined in Property 3 in Def. 17.*

Proof. This follows from a similar argument from previous black-box commit-and-prove literature [IKOS07, GLOV12, GOSV14, LP21]. We provide here a self-contained proof.

We want to show that no non-uniform QPT P^* can commit to a m_{com} for which $\phi(m_{\mathsf{com}}) = 0$ and make V accept with non-negligible probability. There are two cases where $\phi(m_{\mathsf{com}}) = 0$:

1. $m_{\mathsf{com}} = \perp$; **or**
2. m_{com} is in $\{0,1\}^{\ell(\lambda)}$, but $\phi(m_{\mathsf{com}}) = 0$.

(Note that Steps 3 to 5 constitute a coin-flipping protocol such that the resulting T must be a size-t pseudo-random subset of $[n]$ even if P^* is malicious. We henceforth assume w.l.o.g. that T is a size-t random subset of $[n]$.)

For Case 1: If this case happens, we know that the $\{\mathsf{v}_i\}_{i\in[n]}$ statistically-bounded in Step 2 are such that $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1,\ldots,\mathsf{v}_n) = \perp$. We now define an undirected graph G with n vertices corresponding to the n views $\{\mathsf{v}_i\}_i$. Assign an edge between vertices i and j in G if v_i and v_j are *inconsistent* w.r.t. $\mathsf{VSS}_{\mathsf{Share}}$ execution. We first argue that the *minimum vertex cover* set B of G must have size $\geq t$. To see this, consider an execution of $\mathsf{VSS}_{\mathsf{Share}}$ where the adversary corrupts the set of players in B with $|B| < t$, and behaves in a way that the views of any player P_j, for $j \notin B$, is v_j. Such an execution is obtained by choosing all the messages from $P_j \in B$ to $P_j \notin B$ as in the view v_j; since B is a vertex cover, every pair of views $(\mathsf{v}_i, \mathsf{v}_j)$ with $i,j \in \bar{B}$ are not connected in the graph G and hence consistent. Finally, by the $(n+1,t)$-perfect verifiable-committing of VSS, such a corruption should not influence the output of the honest players in the $\mathsf{VSS}_{\mathsf{Share}}$ stage, which must be $\perp$ (otherwise, $\mathsf{VSS}_{\mathsf{Recon}}(\mathsf{v}_1,\ldots,\mathsf{v}_n) \neq \perp$). This means there will be at least $(n-t)$ views in $\{\mathsf{v}_i\}_{i\in[n]}$ indicating that the $\mathsf{VSS}_{\mathsf{Share}}$ execution fails. Since V checks t out of them randomly, V will learn this information and rejects in Step 7c of the Prove Stage *except* with probability $\leq (t/n)^t$, which is negligible in λ due to our parameter setting. That is, $|B| \geq t$ with overwhelming probability.

Now, recall that V checks the consistency of a size-t random subset of all the views $\{\mathsf{v}_i\}_{i\in[n]}$ (i.e., vertices in G). We only need to argue that such a checking will hit an edge in G with overwhelming probability. For this, we use the well-known connection between the size of a minimum vertex cover to the size of a *maximum matching*. Concretely, the graph G must have a *matching*[35] $\mathcal{M}$ of size at least $t/2$. (Otherwise, if the maximum matching contains less than $t/2$ edges, then the vertices of this matching form a vertex cover set B with $|B| < t$.) Recall that if V hits any edge of G, he will reject. The probability that the t vertices (views) that V picks miss all the edges of G is smaller than the probability that he misses all edges of the matching, which is again at most $2^{-\Omega(t)} = 2^{-\Omega(\lambda)}$. To see that, suppose that the first $t/2$ vertices picked by V do not hit an edge of the matching. Denote this set of vertices as $S_{t/2}$. It follows from Serfling's Inequality (see Lem. 1) that with overwhelming probability over λ, $S_{t/2}$ contains $\Omega(t)$ vertices that are the vertices of the edges $\mathcal{M}$. Then, their $\Omega(t)$ matching neighbors will have $\Omega(t/n) = \Omega(1)$ probability of being hit by each subsequent vertex picked by V. Since V will pick $t/2$ more vertices, the probability that V misses all the $\Omega(t)$ matching neighbors with probability at most $2^{-\Omega(t)} = 2^{-\Omega(\lambda)}$.

For Case 2: In this case, we know that m_{com} does not satisfy ϕ. However, the $\{\mathsf{v}'_i\}_{i\in[n]}$ committed by Com in Step 2 of the Prove Stage are supposed to be the views of n parties executing Π_{MPC}. Then, we

[35] Recall that a matching is a set of edges without common vertices.

can use the same argument as for Case 1 to show that V must rejects except with negligible probability. That is, we define the "inconsistency graph" G corresponding to $\{\mathsf{v}'_i\}_{i\in[n]}$, and argue either that there are too many inconsistent views (such that V will catch by checking t of them), or that most parties are honest and report $\phi(m_{\mathsf{com}}) = 0$ (such that V will learn this with overwhelming probability). The only change is, we now rely on the the (n,t)-perfect robustness of Π_{MPC}, instead of the perfectly verifiable-committing of VSS. One caveat is that we need to ensure that P use $\{\mathsf{v}_i\}_{i\in[n]}$ from the Commit Stage to execute Π_{MPC} in Step 1 of the Prove Stage. By checking Step 7b, V is convinced with overwhelming probability that this is indeed the case for at least $(n-t)$ views out of n. This suffices to use the above inconsistency-graph argument. Since this argument is almost identical to Case 1, we omit the details. □

Lemma 21 (ε-Zero-Knowledge). *Assume* ExtCom *and* Com *are computationally-hiding,* ExtCom *is weakly extractable with ε-simulation,* VSS *is $(n+1,t)$-secret (see Def. 1), and* Π_{MPC} *is (n,t)-semi-honest computationally private (see Def. 4). Then, Prot. 6 satisfies the ε-zero-knowledge property defined in Property 4 in Def. 17.*

Proof. This proof is also standard in existing black-box commit-and-prove literature. We provide it for completeness.

The Commit Stage simulator $\mathcal{S}_{\mathsf{EC}}$ behaves identically as the honest prover, except that he executes the $\mathsf{VSS}_{\mathsf{Share}}$ of Step 1 (in his head) to secret-share an arbitrary value, say $0^{\ell(\lambda)}$. Note that $\mathcal{S}_{\mathsf{EC}}$ is straight-line and simulates the honest prover with only $\mathsf{negl}(\lambda)$ error (i.e., satisfying Eq. (10)), thanks to the computationally-hiding property of ExtCom.

To define the Prove Stage simulator, first notice that Steps 3 to 5 of the Prove Stage constitute a ε-simulatable coin-flipping protocol against the malicious non-uniform QPT $V^*(\rho)$, because ExtCom is a ε-simulatable weakly-extractable commitment. The Prove Stage simulator $\mathcal{S}_{\mathsf{Pr}}$ works as follows:

1. Sample at random a size-t subset $\widetilde{T} \subseteq [n]$;
2. Invoke the simulator for Π_{MPC} on parties $\{P_i\}_{i\in[T]}$ to obtain the simulated view $\{\widetilde{\mathsf{v}}'_i\}_{i\in[T]}$; set $\widetilde{\mathsf{v}}'_j$ to all-0 strings of proper length for all $j \in [n] \setminus \widetilde{T}$;
3. Commit to $\{\widetilde{\mathsf{v}}'_i\}_{i\in[n]}$ using Com in parallel as Step 2;
4. For Steps 3 to 5, invoke the coin-flipping simulator to force the resulting r such that it will determine the set $\widetilde{T}$ he sampled in the 1st step;
5. Finish the remaining steps as the honest prover.

Because of the computationally-hiding property of Com, $\mathcal{S}_{\mathsf{Pr}}$ is computationally-indistinguishable from the honest prover until Step 2. Then, because of the security of ε-simulatable coin-flipping, at the end of Step 5, $\mathcal{S}_{\mathsf{Pr}}$ is at most ε-far from the honest prover and will force the result to be $\widetilde{T}$ successfully. Finally, the remaining steps are again computationally-indistinguishable from the honest prover, because of the $(n+1,t)$-secret of VSS and the (n,t)-semi-honest computational privacy of Π_{MPC}. In total, $\mathcal{S}_{\mathsf{Pr}}$ is at most ε-computationally distinguishable from the honest prover (i.e., satisfying Eq. (11)). □

7 Black-Box ε-Simulatable PQ-2PC in Constant Rounds

7.1 Definition and Notation

We first present the formal definition for ε-simulatable two-party computation. It is identical to the standard 2PC definition in the classical setting except that:

1. The malicious party can be QPT;

2. The indistinguishability between the real-world execution and the simulated one is parameterized by a noticeable function $\varepsilon(\lambda)$.

Consider two parties P_1 and P_2 with inputs x_1 and x_2 that wish to interact in a protocol Π to evaluate a 2-party functionality f on their joint inputs. The ideal and real executions follow the standard description as in, e.g., [Gol04].

In the real world, a QPT adversary $\mathcal{A}_\lambda$ with a quantum auxiliary input ρ_λ corrupting $P_i(x_i)$ ($i \in \{0,1\}$) interacts with $P_{1-i}(x_{1-i})$. Let $\mathbf{x} = (x_1, x_2)$ denote the inputs to the two parties. Let $\mathsf{REAL}_{\Pi,\mathcal{A},i}(\lambda, \mathbf{x}, \rho_\lambda)$ denote the random variable consisting of the output of the adversary (which may be an arbitrary function of its view and in particular may be a quantum state) and the outputs of the honest party P_{1-i}.

In the ideal world, a QPT machine $\mathcal{S}$ controls the same party P_i as $\mathcal{A}_\lambda$. It gets x_i and ρ_λ as input. Similar as in the ε-ZK definition [CCY21], $\mathcal{S}$ additionally takes as input a "slackness parameter"[36] $\varepsilon(\lambda)$, which is a noticeable function on λ. Henceforth, we always require that $\mathcal{S}$'s running time is a polynomial on both λ and ε^{-1}. Let $\mathsf{IDEAL}_{f,\mathcal{S},i}(\lambda, \varepsilon, \mathbf{x}, \rho_\lambda)$ denote the outputs of $\mathcal{S}$ (with slackness ε) and the uncorrupted party P_{1-i} from the ideal-world execution.

We remark that, throughout this paper, we only focus on *static* adversaries and *security with abortion* (i.e., the ideal-world adversary (aka the simulator) learns the its output first, and then can instruct the ideal functionality to deliver the output to the honest party or not). This is standard in 2PC literature as security without abortion (aka *fairness*) is impossible for *general-purpose* two-party protocols [Cle86, ABMO15].

Definition 25 (Post-Quantum ε-Simulatable 2PC). *Let f be a classical 2-party functionality, and Π be a classical 2-party protocol. We say that Π is a* ε-simulatable *protocol for f if there exists a QPT simulator $\mathcal{S}$ such that for any non-uniform QPT adversary $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, any $i \in \{0,1\}$, any $\mathbf{x} \in (\{0,1\}^*)^2$, and any noticeable function $\varepsilon(\lambda)$, it holds that:*

$$\{\mathsf{REAL}_{\Pi,\mathcal{A},i}(\lambda, \mathbf{x}, \rho_\lambda)\}_{\lambda\in\mathbb{N}} \overset{c}{\approx}_\varepsilon \{\mathsf{IDEAL}_{f,\mathcal{S},i}(\lambda, \varepsilon, \mathbf{x}, \rho_\lambda)\}_{\lambda\in\mathbb{N}}.$$

7.2 Non-Concurrent Composition of Post-Quantum ε-Simulatable Protocols

We now prove a lemma that will allow us to securely compose different ε-simulatable protocols in the post-quantum setting, as long as the composition happens in a "non-concurrent" manner (explained later). It will be used later since our ε-simulatable 2PC construction in Sec. 7.4 relies on such composition. This lemma is a straightforward extension of the non-concurrent composition lemma from [Can00] to the post-quantum ε-simulatable 2PC protocols in Def. 25.

The Hybrid Model. We start by specifying the model for evaluating a 2-party function g with the assistance of a trusted party computing some 2-party functionalities $(f_1, \ldots, f_m)$. The trusted party is invoked at special rounds, determined by the protocol. In each such round, a function f (out of $f_1, \ldots, f_m$) is specified. At this point, both parties pause the execution of π, store their current state, and then start to make the call to the ideal functionality f. Upon receiving the output back from the trusted party, the protocol π continues.

The protocol π is such that f_{i+1} can be called only if the invocation of f_{i+1} is completely finished. It is possible that $f_i = f_j$ for some $i \neq j$, representing that the same ideal functionality is invoked twice at different time point. We emphasize that, during the invocation of some f_i, the honest party does not send/respond any other messages until it finishes the execution with f_i. This

[36] Actually, [CCY21] refers to it as the "accuracy parameter". But we think "slackness" is a better name as ε measures how *far* two ensembles are.

is called "non-concurrent requirement" in [Can00]. For an execution of π in the $(f_1, \ldots, f_m)$-hybrid model, we use $\mathsf{EXEC}^{f_1,\ldots,f_m}_{\pi,\mathcal{A},i}(\lambda, \mathbf{x}, \rho_\lambda)$ to denote the joint outputs of the adversary corrupting $P_i(x_i)$ and the honest $P_{1-i}(x_{1-i})$.

Let $(\rho_1, \ldots, \rho_m)$ be "subroutine" protocols that are supposed to compute $(f_1, \ldots, f_m)$ respectively. We use $\pi^{\rho_1,\ldots,\rho_m}$ to denote the (plain-model) protocol obtained by replacing the ideal calls in the $(f_1, \ldots, f_m)$-hybrid protocol π with the corresponding "subroutine" protocols in $(\rho_1, \ldots, \rho_m)$.

With these notations, we present the composition lemma in Lem. 22.

Lemma 22 (Non-Concurrent Composition of Post-Quantum ε-Simulatable Protocols). *Let $m \in \mathbb{N}$ be a constant. For $i \in [m]$, let ρ_i be a post-quantum ε-simulatable protocol for a 2-party functionality f_i. Let π be a ε-simulatable protocol for a 2-party functionality g in the $(f_1, \ldots, f_m)$-hybrid model* where no more than one ideal evaluation call is made at each round. *Then, $\pi^{\rho_1,\ldots,\rho_m}$ is a ε-simulatable protocol for g.*

Proof. We will show the proof for the case $m = 1$. The proof for $m > 1$ follows straightforwardly by sequentially repeating the following argument for $m = 1$ to replace $(f_1, \ldots, f_m)$ one-by-one, because the calls to each f_i happen sequentially (see also [Can00, Theorem 5 and Corollary 7] for more details).

Since π is a ε-simulatable protocol for g in the f-hybrid model, there is a QPT $\mathcal{S}'$ such that for any non-uniform QPT $\mathcal{A}' = \{\mathcal{A}'_\lambda, \alpha_\lambda\}_{\lambda\in\mathbb{N}}$, any $i \in \{0, 1\}$, any $\mathbf{x} \in (\{0, 1\}^*)^2$, and any noticeable $\varepsilon(\lambda)/2$,

$$\{\mathsf{IDEAL}_{g,\mathcal{S}',i}(\lambda, \frac{\varepsilon}{2}, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}} \overset{\mathrm{c}}{\approx}_{\frac{\varepsilon}{2}} \{\mathsf{EXEC}^{f}_{\pi,\mathcal{A}',i}(\lambda, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}}. \tag{12}$$

Now, for the LHS execution of Eq. (12), consider a new $\mathcal{S}$ that on input ε, it runs $\mathcal{S}'$ with slackness parameter $\varepsilon/2$. Note that $\mathcal{S}$'s running time is also a polynomial on λ and ε^{-1}. Indeed, $\mathcal{S}$ is identical to $\mathcal{S}'$ with only syntactical changes. Thus, we have

$$\{\mathsf{IDEAL}_{g,\mathcal{S},i}(\lambda, \varepsilon, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}} \overset{\text{i.d.}}{\equiv} \{\mathsf{IDEAL}_{g,\mathcal{S}',i}(\lambda, \frac{\varepsilon}{2}, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}} \overset{\mathrm{c}}{\approx}_{\frac{\varepsilon}{2}} \{\mathsf{EXEC}^{f}_{\pi,\mathcal{A}',i}(\lambda, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}}. \tag{13}$$

Therefore, Lem. 22, follows from the following Claim 10. □

Claim 10. *Let ρ, f and π be the same as in Lem. 22. There exists a quantum adversary $\mathcal{A}'$ in the f-hybrid execution of π such that for any non-uniform QPT $\mathcal{A} = \{\mathcal{A}_\lambda, \alpha_\lambda\}_{\lambda\in\mathbb{N}}$ in the real execution of π^ρ, any $i \in \{0, 1\}$, any $\mathbf{x} \in (\{0, 1\}^*)^2$, and any noticeable $\varepsilon(\lambda)$,*

$$\{\mathsf{EXEC}^{f}_{\pi,\mathcal{A}',i}(\lambda, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}} \overset{\mathrm{c}}{\approx}_{\frac{\varepsilon}{2}} \{\mathsf{REAL}_{\pi^\rho,\mathcal{A},i}(\lambda, \mathbf{x}, \alpha_\lambda)\}_{\lambda\in\mathbb{N}}, \tag{14}$$

where $\mathcal{A}'$'s running time is a polynomial on λ and ε^{-1}.

Proof. This proof proceeds as follows:

1. We construct out of $\mathcal{A} = \{\mathcal{A}_\lambda, \alpha_\lambda\}_{\lambda\in\mathbb{N}}$ a QPT real-world adversary $\mathcal{A}^\rho = \{\mathcal{A}^\rho_\lambda, \alpha^\rho_\lambda\}_{\lambda\in\mathbb{N}}$ that operates against protocol ρ as a stand-alone protocol. The security of ρ guarantees that $\mathcal{A}^\rho$ has a QPT simulator $\mathcal{S}^\rho$ such that for any i, $\mathbf{x}$, and $\varepsilon/2$,

$$\{\mathsf{IDEAL}_{f,\mathcal{S}^\rho,i}(\lambda, \frac{\varepsilon}{2}, \mathbf{x}, \alpha^\rho_\lambda)\}_{\lambda\in\mathbb{N}} \overset{\mathrm{c}}{\approx}_{\frac{\varepsilon}{2}} \{\mathsf{REAL}_{\rho,\mathcal{A}^\rho,i}(\lambda, \mathbf{x}, \alpha^\rho_\lambda)\}_{\lambda\in\mathbb{N}}. \tag{15}$$

2. Out of $\mathcal{A}$ and $\mathcal{S}^\rho$, we construct an adversary $\mathcal{A}'$ that operates against protocol π in the f-hybrid model. We then finish the proof by showing that $\mathcal{A}'$ satisfies Eq. (14) and the running time requirement.

The Simulator $\mathcal{S}^\rho$ for Protocol ρ. We provide more details. Intuitively, $\mathcal{A}^\rho$ represents the "segment" of $\mathcal{A}$ that is involved in the execution of ρ. That is, $\mathcal{A}^\rho$ takes a non-uniform quantum auxiliary input $\{\alpha^\rho_\lambda\}_{\lambda\in\mathbb{N}}$. This auxiliary input contains the internal state of $\mathcal{A}$, controlling party $P_i(x_i)$, and executing the protocol π^ρ with the honest party $P_{1-i}(x_{1-i})$ up to round ℓ_ρ, where ρ is invoked. We note that $\{\alpha^\rho_\lambda\}_{\lambda\in\mathbb{N}}$ can be constructed from $\{\alpha_\lambda\}_{\lambda\in\mathbb{N}}$ (i.e., $\mathcal{A}$'s auxiliary input). At the end of its execution of ρ with $P_{1-i}(x_{1-i})$, adversary A^ρ outputs the current state of the simulated $\mathcal{A}$.

By assumption, ρ is a ε-simulatable protocol for f. Thus, there exists a QPT $\mathcal{S}^\rho$ satisfying Eq. (15).

$\mathcal{A}'$ in the f-Hybrid Model. Adversary $\mathcal{A}'$ represents the "segment" of $\mathcal{A}$ that executes π in the f-hybrid model, where $\mathcal{A}$'s execution of ρ is handled by $\mathcal{S}^\rho$. Recall that $\mathcal{A}$ expects to execute the protocol π_ρ, not π in the f-hybrid model. Therefore, we will need $\mathcal{S}^\rho$ to simulate the ρ part for $\mathcal{A}$. Formally, $\mathcal{A}'$ starts by invoking $\mathcal{A}_\lambda$ on auxiliary input α_λ, and follows $\mathcal{A}$'s instructions up to round ℓ_ρ. At this point, $\mathcal{A}$ expects to execute ρ with P_i, whereas $\mathcal{A}'$ invokes the ideal functionality f (of the f-hybrid model). To continue the execution of $\mathcal{A}$, adversary $\mathcal{A}'$ runs $\mathcal{S}^\rho$. For this purpose, $\mathcal{S}^\rho$ is given the auxiliary input, denoted as α^ρ_λ, that describes the current state of $\mathcal{A}$ at round ℓ_ρ. The information from $\mathcal{S}^\rho$'s trusted party is emulated by $\mathcal{A}'$, using $\mathcal{A}'$'s own ideal functionality f. Recall that the output of $\mathcal{S}^\rho$ is a (simulated) internal state of $\mathcal{A}$ at the completion of protocol ρ. Once protocol ρ completes its execution and the parties return to running π, adversary $\mathcal{A}'$ returns to running $\mathcal{A}$ (starting from the state in $\mathcal{S}^\rho$'s output) and follows the instructions of $\mathcal{A}$. When $\mathcal{A}$ terminates, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs.

With $\mathcal{A}'$ defined above, the LHS and RHS executions of Eq. (14) can be divided into the following 3 stages:

1. Both the LHS and RHS executions are identical up to the starting of round ℓ_ρ
2. At round ℓ_ρ, the RHS $\mathcal{A}$ starts executing ρ with the honesty P_{1-i}, whereas the LHS $\mathcal{A}$ (controlled by $\mathcal{A}'$) talks with $\mathcal{S}^\rho$. At the end of this stage, the internal state of $\mathcal{A}$ in the RHS is $\varepsilon/2$-far from the ($\mathcal{S}^\rho$-simulated) internal state of $\mathcal{A}$ (controlled by $\mathcal{A}'$) in the LHS.
3. Then, both the LHS and RHS again proceed identically until the end.

Since both Stages 1 and 3 are polynomial time, Eq. (14) follows from a straightforward reduction to the ε-simulatability of $\mathcal{S}^\rho$ in Stage 2 (by setting the slackness parameter to $\varepsilon/2$).

Finally, we need to argue that $\mathcal{A}'$'s running time is a polynomial on λ and ε'. This is true as $\mathcal{A}'$ does nothing more than running $\mathcal{A}$ and $\mathcal{S}^\rho$, and both of them is QPT on λ and ε. (Though $\mathcal{S}^\rho$ is invoked with slackness $\varepsilon/2$, $(\varepsilon/2)^{-1}$ is also a polynomial on ε^{-1}.) □

7.3 The Classical Compiler and Parallel Commitments and OTs

We start by recalling the black-box constant-round compiler from semi-honest OT to secure (stand-alone) 2PC in the classical setting. Such a compiler works in 2 steps:

1. Given black-box access to any semi-honest bit-OT, it constructs a malicious-secure string-OT (via [HIK+11, CDMW09]). This step incurs only constant overhead on round complexity.
2. Given black-box access to any maliciously-secure string-OT, it constructs a maliciously-secure 2PC protocol (via [IPS08]). This step incurs only constant overhead on round complexity, *assuming the given OT is parallelly-secure* (see the discussion below).

Although the above steps are widely known to the community, there are subtleties *regarding the stand-alone scenario* that we feel obliged to address. This is because the main focus of [CDMW09] and [IPS08] is the UC setting, and thus their implications for *constant-round constructions in the stand-alone setting* has not been thoroughly discussed. Since this is crucial to the current paper, we

provide further clarification in the following. (While all the aforementioned works are for the more general *multi-party* computation, our discuss here will only focus on the 2-party case.)

Two Flavors of the Hybrid Model. We first need to distinguish between two flavors of the $\mathcal{F}$-hybrid model. This model is a helpful tool enabling modular design and composition of 2PC (and in general, MPC) protocols. For example, to design a 2PC protocol implementing some ideal functionality $\mathcal{G}$ in the $\mathcal{F}$-hybrid model, one can assume that both parties have access to the ideal functionality $\mathcal{F}$. The resulting protocol will be denoted as $\pi^{\mathcal{F}}$, and it will lead to a secure implementation of $\mathcal{G}$ in the plain model if the parties replace the calls to the ideal functionality $\mathcal{F}$ with a protocol ϕ which securely implements $\mathcal{F}$, as long as they do not interleave other parts of $\pi^{(\cdot)}$ with the execution of ϕ.

The meaning of $\mathcal{F}$-hybrid model may vary depending on whether the concerned security is in the stand-alone model or the UC model. In the stand-alone model (e.g., [Can00]), the $\mathcal{F}$-hybrid model only allows the parties to make a single call of the ideal $\mathcal{F}$, because stand-alone protocols may not be concurrently (or even parallelly) composable. Traditionally, if n calls to the idea functionality is needed, people denote this hybrid model as the $(\mathcal{F}_1, \ldots, \mathcal{F}_n)$-hybrid model, and the protocol in this model as $\pi^{\mathcal{F}_1,\ldots,\mathcal{F}_n}$, even if all the $\mathcal{F}_i$'s are actually the same functionality. Importantly, no calls to $\mathcal{F}_i$ (or other parts of $\pi^{(\cdot)}$) are allowed when some $\mathcal{F}_j$ is running. Such composition is called "non-concurrent composition" in [Can00]. To distinguish with the one that we will discuss next, we refer to this model as the *stand-alone* $\mathcal{F}$-hybrid model.

In the UC model ([Can01]), the ideal functionality is extended by including a session ID, and multiple calls to the same functionality will be distinguished by different session IDs. Moreover, the strong composability of the UC security allows us to schedule the calls to $\mathcal{F}$ arbitrarily when designing $\pi^{\mathcal{F}}$ in the $\mathcal{F}$-hybrid model. In particular, parties executing $\pi^{\mathcal{F}}$ can make multiple calls to the ideal $\mathcal{F}$ in parallel (or interleaved arbitrarily), and these calls can also be interleaved with other parts of $\pi^{(\cdot)}$; moreover, parties executing a protocol $\pi^{\mathcal{F}_1,\mathcal{F}_2}$ can even interleave their calls to the two distinct ideal functionalities if necessary. We refer to this hybrid model as the *UC* $\mathcal{F}$-hybrid model. It is worth noting that if we want to replace the ideal $\mathcal{F}$ call(s) in the UC-secure protocol $\pi^{\mathcal{F}}$ (in the UC $\mathcal{F}$-hybrid model), we must use a protocol ϕ that *UC-securely realizes* $\mathcal{F}$. In particular, if ϕ only securely implements $\mathcal{F}$ in the stand-alone sense, no security (not even stand-alone security) is guaranteed for the resulting protocol.

For Step 1. [HIK+11][37] presents the first constant-round black-box compiler from semi-honest bit-OT to maliciously-secure *bit*-OT in the stand-alone setting. This compiler requires a coin-tossing protocol (satisfying the simulation-based security), which can be built from any protocol that implements the ideal commitment functionality $\mathcal{F}_{\text{COM}}$ in the stand-alone setting. It is also worth noting that the security proof in [HIK+11] relies heavily on rewindings, *even when being analyzed in the stand-alone* $\mathcal{F}_{\text{COM}}$*-hybrid model*. Also, at that time, it was unclear if the resulting protocol is parallelly composable (or if it leads to a maliciously-secure *string*-OT).

Later, [CDMW09] shows that [HIK+11] with slight modification yields a black-box constant-round compiler from semi-honest bit-OTs to maliciously-secure *string*-OTs. Moreover, [CDMW09] also simplifies the security proof such that no rewinds are needed in the $\mathcal{F}_{\text{COM}}$-hybrid model. One caveat here is that the "$\mathcal{F}_{\text{COM}}$-hybrid model" in [CDMW09] is different from that in [HIK+11]. The $\mathcal{F}_{\text{COM}}$ in [HIK+11] is just the stand-alone string-commitment functionality. Such an $\mathcal{F}_{\text{COM}}$ suffices for the application of coin-flipping as required by the [HIK+11] compiler. In contrast, the $\mathcal{F}_{\text{COM}}$ employed by the [CDMW09] needs to be a commitment that captures *bounded-parallel security with selectively-opening*. That is, it can be used by a committer to commit to an a-priori bounded number,

[37] This paper is the journal version merging two previous works [IKLP06, Hai08].

say a polynomial $t(\lambda)$, of strings by a single invocation; later, the receiver can specify an arbitrary subset $T \subseteq [t]$ of positions, and the committer will decommit to the i-th commitment for all $i \in T$. More accurately, we denote this ideal functionality as $\mathcal{F}_{\text{SO-COM}}^{t}$ and present it in Fig. 1.[38]

Figure 1: The Ideal Functionality $\mathcal{F}_{\text{SO-COM}}^{t}$

Commit Stage: $\mathcal{F}_{\text{SO-COM}}^{t}$ receives from the committer C a query $\big(\mathsf{Commit}, sid, (m_1, \ldots, m_t)\big)$. $\mathcal{F}_{\text{SO-COM}}^{t}$ records $\big(sid, (m_1, \ldots, m_t)\big)$ and sends $(\mathsf{Receipt}, sid)$ to the receiver R. $\mathcal{F}_{\text{SO-COM}}^{t}$ ignores further Commit messages with the same sid.

Decommit Stage: $\mathcal{F}_{\text{SO-COM}}^{t}$ receives from R a query $(\mathsf{Reveal}, sid, I)$, where I is a subset of $[t]$. If no $\big(sid, (m_1, \ldots, m_t)\big)$ has been recorded, $\mathcal{F}_{\text{SO-COM}}^{t}$ does nothing; otherwise, it sends to R the message $\big(\mathsf{Open}, sid, \{m_i\}_{i \in I}\big)$.

The reason why [CDMW09] needs such an $\mathcal{F}_{\text{SO-COM}}^{t}$ is that their compiler relies on the "cut-and-choose" technique where a party first commits to polynomially-many random strings *in parallel*, and later opens a subset of them (determined by a coin-tossing) for the other party to check. We emphasize that these commitments must be done in parallel; otherwise (i.e., when done sequentially), the resulting protocol will not be in constant rounds. In summary, [CDMW09] actually proved the following lemma.

Lemma 23 ([CDMW09, Proposition 1]). *There is a polynomial $t(\lambda)$ such that there exists a black-box construction of a string-OT protocol secure against static, malicious adversaries in the stand-alone $\mathcal{F}_{\text{SO-COM}}^{t}$-hybrid model (or alternatively, the UC $\mathcal{F}_{\text{COM}}$-hybrid model), starting from any bit-OT protocol secure against static, semi-honest adversaries. Moreover, the construction achieves a constant multiplicative blow up in the number of rounds, and has a strictly polynomial-time and straight-line simulator.*

For Step 2. We now discuss the [IPS08] compiler from OTs to 2PC. As mentioned above, [IPS08] mainly focuses on the UC setting. It proves that in the UC $\mathcal{F}_{\text{OT}}$-hybrid model, there is a constant-round black-box protocol $\Pi_{\text{2PC}}^{\mathcal{F}_{\text{OT}}}$ of general-purpose UC-secure 2PC. As mentioned earlier, it is *in general* unclear what would happen if the $\mathcal{F}_{\text{OT}}$ in $\Pi_{\text{2PC}}^{\mathcal{F}_{\text{OT}}}$ is replaced by a stand-alone secure OT protocol. However, for the special case of [IPS08], it is known that the two participants of the $\Pi_{\text{2PC}}^{\mathcal{F}_{\text{OT}}}$ protocol only make parallel calls to $\mathcal{F}_{\text{OT}}$ *for an a-priori bounded number* t,[39] which is a polynomial on λ. Therefore, similar as the above $\mathcal{F}_{\text{SO-COM}}^{t}$, we can also define a bounded-parallel OT functionality $\mathcal{F}_{\text{OT}}^{t}$ (in Fig. 2) in the stand-alone setting, and interpret the [IPS08] compiler in the stand-alone setting. This leads to following special case of the [IPS08] result (see also Rmk. 9).

Lemma 24 (Special Case of [IPS08, Theorem 3]). *There exists an a-priori known polynomial $t(\lambda)$ such that in the stand-alone $\mathcal{F}_{\text{OT}}^{t}$-hybrid model (or alternatively, the UC $\mathcal{F}_{\text{OT}}$-hybrid model), there is a general-purpose 2PC protocol that achieves stand-alone security against static, malicious adversaries. Moreover, the protocol is constant-round and has a strictly polynomial-time and straight-line simulator.*

Remark 9. Alternatively, one can also replace the t parallel-OT instances with t sequential stand-alone secure OT instances. While this will indeed lead to a secure 2PC protocol in the stand-alone setting, the resulting protocol will not be constant-round as $t = \Omega(\lambda)$ for the [IPS08] construction. Another caveat here is that in a larger protocol where parallel OTs are used, replacing these parallel OTs with sequential OTs may jeopardize security. Nevertheless, there exist standard techniques to achieve the same effect of t parallel OTs using t sequential executions of random OTs.

[38] We note that this $\mathcal{F}_{\text{SO-COM}}^{t}$ has recently been formalized in [GLSV21].

[39] This has been observed and employed in earlier works (e.g., [PW09, Wee10, Goy11]). Thus, we suppress further explanation and refer the reader to [IPS08].

Figure 2: The Ideal Functionality $\mathcal{F}_{\text{OT}}^t$

Sender's Message: $\mathcal{F}_{\text{OT}}^t$ receives from the sender S a query $\left(\mathsf{Send}, sid, \{(x_0^i, x_1^i)\}_{i\in[t]}\right)$. $\mathcal{F}_{\text{OT}}^t$ records $\left(sid, \{(x_0^i, x_1^i)\}_{i\in[t]}\right)$. $\mathcal{F}_{\text{OT}}^t$ ignores further Send messages with the same sid.

Receiver's Message: $\mathcal{F}_{\text{OT}}^t$ receives from the receiver R a query $\left(\mathsf{Receive}, sid, c \in \{0,1\}^t\right)$. If no $\left(sid, \{(x_0^i, x_1^i)\}_{i\in[t]}\right)$ has been recorded, $\mathcal{F}_{\text{OT}}^t$ does nothing; otherwise, it sends to R the message $\left(\mathsf{Open}, sid, \{x_{c_i}^i\}_{i\in[t]}\right)$, where c_i is the i-th bit of c.

The Final Compiler (Classical). Let us summarize the semi-honest bit-OTs to 2PC compiler in the classical setting. First, because of Lem. 24, it suffices to build a protocol that realizes the $\mathcal{F}_{\text{OT}}^t$ functionality w.r.t. stand-alone security, where t is some a-priori known polynomial on λ. To do that, one may want to use Lem. 23. However, Lem. 23 only securely implements the *stand-alone* $\mathcal{F}_{\text{OT}}$, instead of $\mathcal{F}_{\text{OT}}^t$ (i.e., the t-parallel version of $\mathcal{F}_{\text{OT}}$) as required by Lem. 24. To address this issue, we need the following observation regarding parallel composability in the (stand-alone) hybrid model.

Lemma 25 ([GLSV21, Theorem 3.3]). *Assume we have a protocol π that securely realizes an ideal functionality $\mathcal{G}$ in the stand-alone $\mathcal{F}$-hybrid model and* with a straight-line simulator. *Then, a parallel repetition of π (denoted as $\pi^{||}$) implements $\mathcal{G}^{||}$ in the stand-alone $\mathcal{F}^{||}$-hybrid model, where $\mathcal{G}^{||}$ and $\mathcal{F}^{||}$ are the parallel version of $\mathcal{G}$ and $\mathcal{F}$ respectively.*

Lem. 25 had been a folklore and was recently formally proven in [GLSV21]. The intuition behind it is that since the simulator for π is straight-line, the advantage of the simulator (i.e., the ability to extract malicious parties' "secrets") must come from the fact that it emulates the ideal $\mathcal{F}$ for the malicious parties (recall that we are in the $\mathcal{F}$-hybrid model). Therefore, when π is executed in parallel in the $\mathcal{F}^{||}$-hybrid model, the simulator can extract the "secrets" for all the sessions by emulating $\mathcal{F}^{||}$, which allows it to finish the simulation as other parts of the simulation are straight-line. We emphasize that [GLSV21] proved Lem. 25 *in the post-quantum setting* (i.e., it considers classical protocols but requires security against QPT adversaries).

We can combine Lem. 23 and Lem. 25 to obtain a secure implementation of $\mathcal{F}_{\text{OT}}^t$—Observe that the simulator in Lem. 23 is straight-line; therefore, according to Lem. 25, t-parallel repetition of the Lem. 23 compiler will lead to a secure implementation of the desired $\mathcal{F}_{\text{OT}}^t$ in the stand-alone $(\mathcal{F}_{\text{SO-COM}}^t)^{||_t}$-hybrid model, where $(\mathcal{F}_{\text{SO-COM}}^t)^{||_t}$ is the *t-parallel version of* the functionality $\mathcal{F}_{\text{SO-COM}}^t$. Moreover, notice that $(\mathcal{F}_{\text{SO-COM}}^t)^{||_t}$ can be recast as $\mathcal{F}_{\text{SO-COM}}^{t'}$ with a different t', which is also an a-priori known polynomial on λ.

In summary, we obtain the following Thm. 11 by combining Lem. 23 to 25. It is worth noting that previous works claiming constant-round black-box *stand-alone* secure 2PC/MPC from semi-honest OTs follows (albeit implicitly sometimes) this recipe [PW09, Wee10, CDMW09, Goy11].

Theorem 11. *There is a polynomial $t'(\lambda)$ such that there exists a black-box construction of 2PC protocol secure against static, malicious adversaries in the stand-alone $\mathcal{F}_{\text{SO-COM}}^{t'}$-hybrid model, starting from any bit-OT protocol secure against static, semi-honest adversaries. Moreover, the construction achieves a constant multiplicative blow up in the number of rounds, and has a strictly polynomial-time and straight-line simulator.*

7.4 Our Construction of ε-Simulatable Post-Quantum 2PC

Now we are ready to establish the following theorem:

Theorem 12. *Assuming the existence of a constant-round semi-honest bit-OT secure against QPT adversaries, there exists a black-box, constant-round construction of ε-simulatable 2PC protocol secure against QPT adversaries.*

To prove Thm. 12, we follows the approach shown in Sec. 7.3. There are two key differences that require special attention:

1. The recipe from Sec. 7.3 gives 2PC secure against (classical) PPT adversaries; but we want to to achieve security against QPT adversaries.
2. We only require ε-simulatable security. That is, for any noticeable function $\varepsilon(\lambda)$, the simulator will generate a view for the corrupted party that is at most ε-far from that party's view in the real-world execution; the running time of the simulator should be a polynomial on both the security parameter and $1/\varepsilon$ (Def. 25). We remark that ε-simulatable security is effectively the best one can hope for, because constant-round post-quantum 2PC satisfying the standard $\mathsf{negl}(\lambda)$-close simulation cannot exist, unless either *non-black-box simulation* is used to prove security or $\mathbf{NP} \subseteq \mathbf{BQP}$ [CCLY21].

To deal with these issues, we first note that the classical recipe from Sec. 7.3 actually extends to the post-quantum setting (i.e., when the adversaries are QPT), because the simulators in both Lem. 23 and 24 are straight-line and non-cloning (this has also been observed in previous works, e.g., [Unr10, ABG+21b]); and as mentioned earlier, Lem. 25 was originally proven in the post-quantum setting directly. Therefore, it seems that we immediately obtain a post-quantum version of Thm. 11. But there is one more caveat—Thm. 11 also relies on the aforementioned non-concurrent composition lemma. In particular, the $\mathcal{F}_{\text{OT}}^{t}$ functionality assumed in Lem. 24 is realized by the real-world protocol (albeit in the $F_{\text{SO-COM}}^{t}$-hybrid model) induced by Lem. 23 (together with Lem. 25). Therefore, we need to make sure that such a non-concurrent composition is also applicable in the post-quantum setting. Fortunately, this follows from Lem. 22 which we proved in Sec. 7.2. The above discussion leads to the following post-quantum and ε-simulatable version of Thm. 11:

Theorem 13. *There is a polynomial $t(\lambda)$ such that there exists a black-box construction of ε-simulatable 2PC protocol secure against static, malicious QPT adversaries in the stand-alone $\mathcal{F}_{\text{SO-COM}}^{t}$-hybrid model, starting from any bit-OT protocol secure against static, semi-honest QPT adversaries. Moreover, the construction achieves a constant multiplicative blow up in the number of rounds.*

Now, to finish the proof of Thm. 12, we only need to show the following Lem. 26, and then invoke the non-concurrent composition lemma (Lem. 22) again, to replace the $\mathcal{F}_{\text{SO-COM}}^{t}$ in Thm. 13 with the semi-honest OT based construction from Lem. 26.

Lemma 26. *For any polynomial $t(\lambda)$, there exists a constant-round protocol that implements $\mathcal{F}_{\text{SO-COM}}^{t}$ w.r.t. ε-simulatable security against QPT adversaries. This construction makes only black-box access to any OWFs secure against QPT adversaries*[40].

Proof. We show a ε-simulatable protocol implementing $\mathcal{F}_{\text{SO-COM}}^{t}$, assuming only black-box access to OWFs secure against QPT adversaries. We will again rely on the black-box ExtCom-and-Prove as per Def. 17. The committer will commit to the t messages $(m_1, \ldots, m_t)$ using the Commit Stage of the ExtCom-and-Prove. To decommit to the messages determined by the receiver's choice of $I \subseteq [n]$, the committer first sends $\{m_i\}_{i\in I}$ to the receiver; then, both parties execute the Prove Stage, where the committer proves the following predicate:

$$\phi_{I,\{m_i\}_{i\in I}}(x) = \begin{cases} 1 & \text{if } (x = m'_1 \| \ldots \| m'_t) \wedge (\forall i \in I, m'_i = m_i) \\ 0 & \text{otherwise} \end{cases}. \tag{16}$$

[40] Note that such OWFs can be constructed from any post-quantum semi-honest OT in black-box.

That is, $\phi_{I,\{m_i\}_{i\in I}}(\cdot)$ has I and $|I|$-many messages hard-wired; it evaluates to 1 if and only if its input can be parsed as $(m'_1\|\ldots,\|m'_t)$ of the correct length, and m'_i agrees with m_i for all i's specified by I.

This construction is constant-round and based on black-box access to post-quantum secure OWFs, because the ExtCom-and-Prove scheme is so. To prove security, if the committer is corrupted, simulation can be done via the ε-simulatable extractability of the Commit Stage (Property 1); if the receiver is corrupted, simulation can be done via the ε-ZK property (Property 4) of the ExtCom-and-Prove. □

8 Black-Box Constant-Round ε-2PC using Quantum Communication

If quantum communication is allowed, we can obtain a constant-round ε-simulatable PQ-2PC assuming only black-box access to post-quantum secure OWFs (instead of post-quantum secure semi-honest OTs). Recently, [GLSV21] and [BCKM21] (based on earlier works [CK90, BBCS92, DFL+09, BF10]) constructed fully simulatable (in contrast to ε-simulatable) post-quantum 2PC (actually, they obtained MPC) assuming only post-quantum secure OWFs and quantum communication. The protocol from [BCKM21] makes only black-box use of the assumed OWF. But neither of these constructions is in constant rounds (without trusted assumptions like common reference strings).

Our construction follows the same path shown in Sec. 7.4. That is, we break the task of constructing ε-simulatable PQ-2PC in to the following steps:

1. Construct a constant-round ε-simulatable protocol implementing $\mathcal{F}^{t'}_{\text{SO-COM}}$;
2. In the $\mathcal{F}^{t'}_{\text{SO-COM}}$-hybrid model, construct a constant-round ε-simulatable protocol implementing $\mathcal{F}^{t}_{\text{OT}}$;
3. In the $\mathcal{F}^{t}_{\text{OT}}$-hybrid model, construct a constant-round ε-simulatable PQ-2PC protocol.

Notice that in Sec. 7 where only classical communication is allowed, the only place where the post-quantum semi-honest OT is used is Step 2. (Steps 1 and 3 only need to make black-box access to a post-quantum secure OWF.) Therefore, we will obtain the desired construction if we can make Step 2 work without relying on post-quantum secure semi-honest OTs.

We observe that [GLSV21] already did this relying on the [BBCS92] OT construction, which makes use of quantum communication. We refer the read to [GLSV21] for further details. Here, we simply import the related lemma from [GLSV21].

Lemma 27 ([GLSV21, Theorem 3.2, Corollary 3.4]). *For any polynomial $t(\lambda)$, there exists another polynomial $t'(\lambda)$ such that there is a constant-round protocol implementing $\mathcal{F}^{t}_{\text{OT}}$ w.r.t. ε-simulatable security against QPT adversaries in the $\mathcal{F}^{t'}_{\text{SO-COM}}$-hybrid model. This protocol makes use of quantum communication.*

Replacing Step 2 with the construction promised by Lem. 27 yields our final theorem. (Recall that we already finished Steps 1 and 3 in Sec. 7.)

Theorem 14. *Assuming the existence of OWFs secure against QPT adversaries, there exists a black-box, constant-round construction of ε-simulatable 2PC protocol secure against QPT adversaries. This protocol makes use of quantum communication.*

Acknowledgments

We thank Susumu Kiyoshima for answering questions regarding the strongly extractable commitment in [Kiy14]. We also thank Omkant Pandey for helpful discussions about the black-box constant-round compiler from semi-honest OTs to 2PC. We thank a reviewer of STOC 2022 for suggesting a simplified proof of Lem. 8 with a better bound.

References

Aar05. Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005. 10

ABG+21a. Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 435–464. Springer, Heidelberg, October 2021. 2, 4

ABG+21b. Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 435–464. Springer, 2021. 55

ABMO15. Gilad Asharov, Amos Beimel, Nikolaos Makriyannis, and Eran Omri. Complete characterization of fairness in secure two-party computation of Boolean functions. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 199–228. Springer, Heidelberg, March 2015. 49

ACP21. Prabhanjan Ananth, Kai-Min Chung, and Rolando L La Placa. On the concurrent composition of quantum zero-knowledge. In *Annual International Cryptology Conference*, pages 346–374. Springer, 2021. 4

AL20. Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 123–152. Springer, Heidelberg, November 2020. 1

BBCS92. Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 351–366. Springer, Heidelberg, August 1992. 1, 56

BCKM21. James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 467–496, Virtual Event, August 2021. Springer, Heidelberg. 1, 3, 4, 56

BF10. Niek J. Bouman and Serge Fehr. Sampling in a quantum population, and applications. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 724–741. Springer, Heidelberg, August 2010. 1, 14, 15, 56

BG20. Anne Broadbent and Alex B. Grilo. QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In *61st FOCS*, pages 196–205. IEEE Computer Society Press, November 2020. 4, 42, 43

BGW88. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988. 15, 16, 17, 33, 45

BJSW20. Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. *SIAM J. Comput.*, 49(2):245–283, 2020. 4

BKP18. Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 671–684. ACM Press, June 2018. 2

BLS21. Nir Bitansky, Huijia Lin, and Omri Shmueli. Non-malleable commitments against quantum attacks. Cryptology ePrint Archive, Report 2021/920, 2021. https://ia.cr/2021/920. 1, 2, 3, 5, 19

Blu86. Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, page 1444–1451, 1986. 61

BS20. Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 269–279. ACM Press, June 2020. 1, 2, 3, 4, 9, 14, 15, 19, 21, 42, 43, 44

BY20. Zvika Brakerski and Henry Yuen. Quantum garbled circuits, 2020. 4

Can00. Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, January 2000. 12, 13, 49, 50, 52

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001. 13, 52

CCLY21. Nai-Hui Chia, Kai-Min Chung, Qipeng Liu, and Takashi Yamakawa. On the impossibility of post-quantum black-box zero-knowledge in constant rounds. 2021. 2, 5, 55

CCY21. Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. A black-box approach to post-quantum zero-knowledge in constant rounds. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume

12825 of *LNCS*, pages 315–345, Virtual Event, August 2021. Springer, Heidelberg. 1, 2, 3, 4, 6, 9, 10, 14, 19, 21, 22, 23, 24, 26, 49, 61, 62

CDD+99. Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations secure against an adaptive adversary. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 311–326. Springer, Heidelberg, May 1999. 15, 33, 45

CDMW09. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 387–402. Springer, Heidelberg, March 2009. 1, 2, 12, 13, 51, 52, 53, 54

CF01. Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–40. Springer, Heidelberg, August 2001. 40

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th FOCS*, pages 383–395. IEEE Computer Society Press, October 1985. 15

CK88. Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *29th FOCS*, pages 42–52. IEEE Computer Society Press, October 1988. 1

CK90. Claude Crépeau and Joe Kilian. Weakening security assumptions and oblivious transfer (abstract). In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 2–7. Springer, Heidelberg, August 1990. 1, 56

Cle86. Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *18th ACM STOC*, pages 364–369. ACM Press, May 1986. 49

CLP20. Rohit Chatterjee, Xiao Liang, and Omkant Pandey. Improved black-box constructions of composable secure computation. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *ICALP 2020*, volume 168 of *LIPIcs*, pages 28:1–28:20. Schloss Dagstuhl, July 2020. 1, 38

DDN00. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000. 1, 7, 28

DFL+09. Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 408–427. Springer, Heidelberg, August 2009. 1, 56

DGJ+20. Yfke Dulek, Alex B. Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multiparty quantum computation with a dishonest majority. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 729–758. Springer, Heidelberg, May 2020. 1

DI05. Ivan Damgård and Yuval Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 378–394. Springer, Heidelberg, August 2005. 1

DNS04. Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004. 2

FGJ18. Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. On the existence of three round zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 3–33. Springer, Heidelberg, April / May 2018. 2

GGJS12. Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Concurrently secure computation in constant rounds. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 99–116. Springer, Heidelberg, April 2012. 8

GGMP16. Sanjam Garg, Divya Gupta, Peihan Miao, and Omkant Pandey. Secure multiparty RAM computation in constant rounds. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part I*, volume 9985 of *LNCS*, pages 491–520. Springer, Heidelberg, October / November 2016. 1

GK96. Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996. 6

GKLW21. Rachit Garg, Dakshita Khurana, George Lu, and Brent Waters. Black-box non-interactive non-malleable commitments. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 159–185. Springer, Heidelberg, October 2021. 1

GKP18. Sanjam Garg, Susumu Kiyoshima, and Omkant Pandey. A new approach to black-box concurrent secure computation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 566–599. Springer, Heidelberg, April / May 2018. 1

GLM+04. Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 258–277. Springer, Heidelberg, February 2004. 1

GLOV12. Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd FOCS*, pages 51–60. IEEE Computer Society Press, October 2012. 1, 3, 8, 11, 18, 36, 47

GLP+15. Vipul Goyal, Huijia Lin, Omkant Pandey, Rafael Pass, and Amit Sahai. Round-efficient concurrently composable secure computation via a robust extraction lemma. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 260–289. Springer, Heidelberg, March 2015. 1

GLPV20. Sanjam Garg, Xiao Liang, Omkant Pandey, and Ivan Visconti. Black-box constructions of bounded-concurrent secure computation. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 87–107. Springer, Heidelberg, September 2020. 1

GLSV21. Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Heidelberg, October 2021. 1, 3, 4, 12, 13, 53, 54, 56

Gol04. Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004. 49

GOSV14. Vipul Goyal, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Black-box non-black-box zero knowledge. In David B. Shmoys, editor, *46th ACM STOC*, pages 515–524. ACM Press, May / June 2014. 1, 3, 36, 47

Goy11. Vipul Goyal. Constant round non-malleable protocols using one way functions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 695–704. ACM Press, June 2011. 1, 2, 9, 12, 53, 54

Hai08. Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 412–426. Springer, Heidelberg, March 2008. 1, 52

HIK+11. Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011. 51, 52

HSS11. Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 411–428. Springer, Heidelberg, August 2011. 1, 2, 3, 4

HV16. Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 397–429. Springer, Heidelberg, August 2016. 1

HV18. Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. Round-optimal fully black-box zero-knowledge arguments from one-way permutations. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 263–285. Springer, Heidelberg, November 2018. 3

IKLP06. Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions for secure computation. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 99–108. ACM Press, May 2006. 1, 52

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007. 1, 3, 11, 17, 18, 36, 37, 47

IKOS08. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 433–442. ACM Press, May 2008. 9

IPS08. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, Heidelberg, August 2008. 1, 12, 13, 51, 53

Kil88. Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988. 1

Kiy14. Susumu Kiyoshima. Round-efficient black-box construction of composable multi-party computation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 351–368. Springer, Heidelberg, August 2014. 1, 8, 56

Kiy20. Susumu Kiyoshima. Round-optimal black-box commit-and-prove with succinct communication. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 533–561. Springer, Heidelberg, August 2020. 3

KOS18. Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. Round optimal black-box "commit-and-prove". In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 286–313. Springer, Heidelberg, November 2018. 1, 3

Lin13. Yehuda Lindell. A note on constant-round zero-knowledge proofs of knowledge. *Journal of Cryptology*, 26(4):638–654, October 2013. 1

LMS21. Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited (or: How to do quantum rewinding undetectably). Cryptology ePrint Archive, Report 2021/1543, 2021. https://ia.cr/2021/1543. 2, 5, 6

LN11. Carolin Lunemann and Jesper Buus Nielsen. Fully simulatable quantum-secure coin-flipping and applications. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 21–40. Springer, Heidelberg, July 2011. 1, 2, 3, 4

LP12. Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 461–478. Springer, Heidelberg, August 2012. 1

LP21. Xiao Liang and Omkant Pandey. Towards a unified approach to black-box constructions of zero-knowledge proofs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 34–64, Virtual Event, August 2021. Springer, Heidelberg. 1, 36, 38, 47

MOSV06. Daniele Micciancio, Shien Jin Ong, Amit Sahai, and Salil P. Vadhan. Concurrent zero knowledge without complexity assumptions. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 1–20. Springer, Heidelberg, March 2006. 1

Nao91. Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991. 7

PRS02. Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd FOCS*, pages 366–375. IEEE Computer Society Press, November 2002. 1, 7, 28

PW09. Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 403–418. Springer, Heidelberg, March 2009. 1, 2, 3, 4, 7, 12, 28, 40, 53, 54

Ros04. Alon Rosen. A note on constant-round zero-knowledge proofs for NP. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 191–202. Springer, Heidelberg, February 2004. 1, 7, 28

Ser74. Robert J Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48, 1974. 15

Unr10. Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 486–505. Springer, Heidelberg, May / June 2010. 55

Unr12. Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012. 4, 7, 14, 15

Wat09. John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. 7, 8, 9, 15

Wee10. Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51st FOCS*, pages 531–540. IEEE Computer Society Press, October 2010. 53, 54

Wyn75. Aaron D Wyner. The wire-tap channel. *Bell system technical journal*, 54(8):1355–1387, 1975. 1

Appendix

A From Extractable Commitment to ZK Argument

We give a proof sketch for that constant-round post-quantum extractable commitment with black-box extraction implies constant-round post-quaautum ZK argument for **NP**. The construction also relies on Σ-protocol (say, Blum's Hamiltonicity protocol [Blu86]). Then the ZK argument works as follows:

1. The prover sends the first message of the Σ-protocol.
2. The prover and verifier jointly run one-sided simulation coin-flipping based on the extractable commitment to determine the second message of Σ-protocol. That is, they do the following:
   (a) The verifier commits to a uniformly random string r_1 of the same length as the second message of Σ-protocol by using the extractable commitment.
   (b) The prover sends a uniformly random string r_2 of the same length in the clear.
   (c) The verifier opens r_1, and the second message of the Σ-protocol is set to be $r_1 \oplus r_2$.
3. The prover sends the third message of the Σ-protocol.
4. The verifier runs the verification algorithm of the Σ-protocol.

It is clear that the above protocol is constant-round if the extractable commitment is constant-round. For soundness, we show that no QPT cheating prover can bias $r_1 \oplus r_2$ based on the computational hiding of the extractable commitment. Then, the soundness of the above protocol can be easily reduced to that of the underlying Σ-protocol. For ZK, recall that Σ-protocol satisfies a special honest-verifier ZK, which enables one to simulate the transcript if the second message is known in advance. Based on that, we construct a simulator for the above protocol as follows. It first randomly chooses the second message β of the Σ-protocol. Then it simulates the transcript (α, β, γ) of the Σ-protocol and sends α to the malicious verifier as the first message. Then it extracts r_1 from the malicious verifier while simulating its state by running the extractor and sets $r_2 := r_1 \oplus \beta$ and sends r_2 to the malicious verifier. Finally, it sends γ to the malicious verifier as the final message. It is straightforward to show that this simulator satisfies the requirement for ZK. Moreover, this simulator is black-box as long as the extractor is black-box.

B Postponed Proofs in Sec. 4

B.1 Proof of Lem. 10

Proof of Lem. 10. Since the proof is very similar to that of [CCY21, Lemma 3.3], we only explain the differences. We define projections Π_0 and Π_1 over $\mathcal{H} = \mathcal{H}_{\mathbf{X}} \times \mathcal{H}_{\mathbf{Y}}$ as

$$\Pi_0 := I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}}, \Pi_1 := \Pi,$$

and apply Jordan's lemma similarly to in the proof of [CCY21, Lemma 3.2]. That is, $S_{\geq\delta}$ (resp. $S_{<\delta}$) is defined to be the subspace spanned by eigenvectors of $\Pi_0\Pi_1\Pi_0$ with eigenvalues $\geq \delta$ (resp. $< \delta$). Then Items 1 and 2 directly follow from Jordan's lemma. For proving Items 3 and 4, we slightly modify the definitions of $U_{\mathsf{amp},T}$ and Amp from those in [CCY21, Lemma 3.2]. We first consider an algorithm $\widetilde{\mathsf{Amp}}$ described as follows:

$\widetilde{\mathsf{Amp}}(1^T, |\psi\rangle_{\mathbf{X},\mathbf{Y}})$: This algorithm takes a repetition parameter T and a quantum state $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in \mathcal{H}$ as input and works as follows:[41]

1. Repeat the following for $i = 1, ..., T$:
   (a) Perform a measurement $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}} - \Pi_0\}$. If the outcome is 1, i.e., if Π_0 is applied, then set $A_i := 1$.
   (b) Perform a measurement $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}} - \Pi_1\}$ If the outcome is 1, i.e., if Π_1 is applied, then set $B_i := 1$.
   (c) If $A_i = B_i = 1$, output the state in the registers $(\mathbf{X}, \mathbf{Y})$ and a classical bit $b = 1$ indicating a success and immediately halt.
2. Output the state in the registers $(\mathbf{X}, \mathbf{Y})$ and a classical bit $b = 0$ indicating a failure.

Remark 10. In the proof of [CCY21, Lemma 3.2], $\widetilde{\mathsf{Amp}}$ outputs the state of $(\mathbf{X}, \mathbf{Y})$ as soon as observing $B_i = 1$ for some i, but here, it outputs the state only after observing $A_i = 1$ and $B_i = 1$ for some i. This modification is needed to ensure that the output state is in the span of $\Pi_1\Pi_0$ (rather than in the span of Π_1 as in [CCY21, Lemma 3.2]).

We define an algorithm Amp as a purified version of $\widetilde{\mathsf{Amp}}$. That is, Amp works similarly to $\widetilde{\mathsf{Amp}}$ except that intermediate measurement results are stored in designated registers in $\mathbf{Anc}$ without being measured and the output b is stored in register $\mathbf{B}$. Let $U_{\mathsf{amp},T}$ be the unitary part of $\mathsf{Amp}(1^T, \cdot)$. Then Item 3a and item 4 follows from the definition and Item 3c follows from the fact that $S_{\geq\delta}$ and $S_{<\delta}$ are invariant under Π_0 and Π_1. For proving Item 3b, it suffices to consider the case where $|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}}$ is an eigenvector of $\Pi_0\Pi_1\Pi_0$ with an eigenvalue $t \geq \delta$ by Jordan's lemma as argued in the proof of [CCY21, Lemma 3.2]. In this case, $\{A_i\}$ and $\{B_i\}$ follow the following distribution:

– $\Pr[A_1 = 1] = 1$ and for all $i \geq 2$, the distribution of A_i only depends on B_{i-1} and $\Pr[A_i = B_{i-1}] = t$.
– For $i \geq 1$, the distribution of B_i only depends on A_i and $\Pr[B_i = A_i] = t$.

It suffices to show that for any noticeable $t = t(\lambda)$ and $\nu = \nu(\lambda)$, there is $T = \mathsf{poly}(\lambda)$ such that

$$\Pr[\exists i \in [T]\ A_i = B_i = 1] \geq 1 - \nu. \tag{17}$$

First, note that

$$\Pr[A_1 = B_1 = 1] = t.$$

Thus, if $t \geq 1 - \nu$, then Eq. (17) is satisfied for $T = 1$. Below, we consider the case where $t < 1 - \nu$.

For any $i \geq 1$, we have

$$\Pr[A_{i+1} = B_{i+1} = 1 \mid B_i = 0] = t(1 - t)$$

and

$$\Pr[A_{i+1} = B_{i+1} = 1 \mid B_i = 1] = t^2.$$

Thus, for any positive integer T, we have

$$\Pr[\exists i \in [T]\ A_i = B_i = 1] \geq t + (1 - t)(1 - (1 - \min\{t(1 - t), t^2\})^{T-1}).$$

Since $\min\{t(1 - t), t^2\}$ is noticeable, we can take $T = \mathsf{poly}(\lambda)$ in such a way that $t + (1 - t)(1 - (1 - \min\{t(1 - t), t^2\})^{T-1}) \geq 1 - \nu$. This completes the proof. □

[41] Strictly speaking, we need to consider descriptions of quantum circuits to perform measurements $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}} - \Pi_0\}$ and $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}} - \Pi_1\}$ as part of its input so that we can make the description of $\widetilde{\mathsf{Amp}}$ independent on them. (Looking ahead, this is needed for showing Item 4 in Lemma 10 where Amp is required to be a uniform QPT machine.) We omit to explicitly write them in the input of $\widetilde{\mathsf{Amp}}$ for notational simplicity.

B.2 Proof of Lem. 8

Proof of Lem. 8. If $(12\gamma^{1/2}+2\delta)^{1/2}>1$, then the desired inequality trivially holds. Thus, we assume $(12\gamma^{1/2}+2\delta)^{1/2}\leq 1$ in the rest of the proof.

F can be represented by a projector Π over the input register and auxiliary input register with which F works as follows. F appends the ancillary register initialized to $|0^n\rangle$, performs the measurement $\{\Pi, I-\Pi\}$ on the input and ancillary registers, and outputs the state in a designated output register tracing out all other registers if the state is projected onto Π, and otherwise outputs Fail.

We consider an additional one-qubit register and define

$$|\psi_b\rangle := \sqrt{1-p_b}\,|0\rangle\,|0^m\rangle\,|0^n\rangle + |1\rangle\,\Pi\,|\phi_b\rangle\,|0^n\rangle$$

for $b\in\{0,1\}$ where m is the number of qubits in the register for $|\phi_b\rangle$ and

$$p_b := \|\Pi\,|\phi_b\rangle\,|0^n\rangle\,\|^2.$$

Without loss of generality, we assume $p_0\geq p_1$. It suffices to prove

$$\|\,|\psi_0\rangle\langle\psi_0| - |\psi_1\rangle\langle\psi_1|\,\|_{tr} \leq (12\gamma^{1/2}+2\delta)^{1/2} \tag{18}$$

because a distinguisher that distinguishes $F(|\phi_0\rangle\langle\phi_0|)$ and $F(|\phi_1\rangle\langle\phi_1|)$ can be easily converted into a distinguisher that distinguishes $|\psi_0\rangle$ and $|\psi_1\rangle$ with the same advantage.

We show the following claim.

Claim 15. *The following holds:*

1. $\|\Pi\left|\phi_{b,0}\right\rangle|0^n\rangle\,\|^2\leq\gamma$ *for* $b\in\{0,1\}$.
2. $\|\Pi\left|\phi_{b,1}\right\rangle|0^n\rangle\,\|^2\geq p_b-3\gamma^{1/2}$ *for* $b\in\{0,1\}$.

Proof of Claim 15. Item 1 follows from the definition of Π and the assumption that $\Pr\left[F\left(\frac{|\phi_{b,0}\rangle\langle\phi_{b,0}|}{\||\phi_{b,0}\rangle\|^2}\right)=\mathsf{Fail}\right]\geq 1-\gamma$. Item 2 can be shown as follows:

$$\begin{aligned}
p_b &= \|\Pi\,|\phi_b\rangle\,|0^n\rangle\,\|^2\\
&= \|\Pi\left|\phi_{b,0}\right\rangle|0^n\rangle + \Pi\left|\phi_{b,1}\right\rangle|0^n\rangle\,\|^2\\
&\leq \|\Pi\left|\phi_{b,0}\right\rangle|0^n\rangle\,\|^2 + \|\Pi\left|\phi_{b,1}\right\rangle|0^n\rangle\,\|^2 + 2\|\Pi\left|\phi_{b,0}\right\rangle\|\cdot\|\Pi\left|\phi_{b,1}\right\rangle\|\\
&\leq \|\Pi\left|\phi_{b,1}\right\rangle|0^n\rangle\,\|^2 + 3\gamma^{1/2}
\end{aligned}$$

where the last inequality follows from $\|\Pi\left|\phi_{b,0}\right\rangle|0^n\rangle\,\|^2\leq\|\Pi\left|\phi_{b,0}\right\rangle|0^n\rangle\,\|\leq\gamma^{1/2}$ by Item 1 and $\|\Pi\left|\phi_{b,1}\right\rangle\|\leq 1$. □

We give a lower bound for $|\,\langle\psi_0|\psi_1\rangle\,|$. By the definition of $|\psi_b\rangle$,

$$\begin{aligned}
|\,\langle\psi_0|\psi_1\rangle\,| &= |\sqrt{(1-p_0)(1-p_1)} + \langle\phi_0|\,\langle 0^n|\,\Pi\,|\phi_1\rangle\,|0^n\rangle\,|\\
&= \left|\begin{array}{ll}\sqrt{(1-p_0)(1-p_1)} + \langle\phi_{0,0}|\,\langle 0^n|\,\Pi\,|\phi_{1,0}\rangle\,|0^n\rangle + \langle\phi_{0,0}|\,\langle 0^n|\,\Pi\,|\phi_{1,1}\rangle\,|0^n\rangle\\ \qquad + \langle\phi_{0,1}|\,\langle 0^n|\,\Pi\,|\phi_{1,0}\rangle\,|0^n\rangle + \langle\phi_{0,1}|\,\langle 0^n|\,\Pi\,|\phi_{1,1}\rangle\,|0^n\rangle\end{array}\right|\\
&= \left|\begin{array}{lll}\sqrt{(1-p_0)(1-p_1)} + \langle\phi_{0,0}|\,\langle 0^n|\,\Pi\,|\phi_{1,0}\rangle\,|0^n\rangle & & + \langle\phi_{0,0}|\,\langle 0^n|\,\Pi\,|\phi_{1,1}\rangle\,|0^n\rangle\\ \qquad + \langle\phi_{0,1}|\,\langle 0^n|\,\Pi\,|\phi_{1,0}\rangle\,|0^n\rangle & & + \langle\phi_{0,1}|\,\langle 0^n|\,\Pi\,|\phi_{0,1}\rangle\,|0^n\rangle\\ \qquad + \langle\phi_{0,1}|\,\langle 0^n|\,\Pi(|\phi_{1,1}\rangle - |\phi_{0,1}\rangle)\,|0^n\rangle & & \end{array}\right|\\
&\geq (1-p_0) + \|\Pi\,|\phi_{0,1}\rangle\,|0^n\rangle\,\|^2 - \sum_{(c,d)\in\{(0,0),(0,1),(1,0)\}} \|\Pi\,|\phi_{0,c}\rangle\,|0^n\rangle\|\cdot\|\Pi\left|\phi_{1,d}\right\rangle|0^n\rangle\,\| - \|\,|\phi_{1,1}\rangle - |\phi_{0,1}\rangle\,\|\\
&\geq (1-p_0) + (p_0-3\gamma^{1/2}) - 3\gamma^{1/2} - \delta\\
&= 1-6\gamma^{1/2}-\delta
\end{aligned}$$

where we used the assumption that $p_0 \geq p_1$ in the first inequality and Claim 15 and the assumption that $\| |\phi_{1,1}\rangle - |\phi_{0,1}\rangle \| \leq \delta$ in the second inequality. We note that $1 - 6\gamma^{1/2} - \delta > 0$ since we assume $(12\gamma^{1/2} + 2\delta)^{1/2} \leq 1$.

Then, we have

$$\begin{aligned} \| |\psi_0\rangle \langle\psi_0| - |\psi_1\rangle \langle\psi_1| \|_{tr} &= \sqrt{1 - |\langle\psi_0|\psi_1\rangle|^2} \\ &\leq \sqrt{12\gamma^{1/2} + 2\delta} \end{aligned}$$

This completes the proof of Lem. 8. □