Best-of-Both-Worlds Multiparty Quantum Computation with Publicly Verifiable Identifiable Abort

Kai-Min Chung[1], Mi-Ying (Miryam) Huang[2], Er-Cheng Tang[1,3], and Jiapeng Zhang[2]

[1]Academia Sinica, Taiwan
[2]University of Southern California, United States
[3]University of Washington, United States

October 10, 2023

Abstract

Alon et al. (CRYPTO 2021) introduced a multiparty quantum computation protocol that is secure with identifiable abort (MPQC-SWIA). However, their protocol allows only inside MPQC parties to know the identity of malicious players. This becomes problematic when two groups of people disagree and need a third party, like a jury, to verify who the malicious party is. This issue takes on heightened significance in the quantum setting, given that quantum states may exist in only a single copy. Thus, we emphasize the necessity of a protocol with *publicly verifiable identifiable abort* (PVIA), enabling outside observers with only classical computational power to agree on the identity of the malicious party in case of an abort. However, achieving MPQC with PVIA poses significant challenges due to the no-cloning theorem, and previous works proposed by Mahadev (STOC 2018) and Chung et al. (Eurocrypt 2022) for classical verification of quantum computation fall short.

In this paper, we obtain the first MPQC-PVIA protocol assuming post-quantum oblivious transfer and a classical broadcast channel. The core component of our construction is a new authentication primitive called *auditable quantum authentication* (AQA) that identifies the malicious sender with overwhelming probability. Additionally, we provide the first MPQC protocol with best-of-both-worlds (BoBW) security, which guarantees output delivery with an honest majority and remains secure with abort even if the majority is dishonest. Our best-of-both-worlds MPQC protocol also satisfies PVIA upon abort.

Contents

1 Introduction

Secure multiparty computation (MPC) allows two or more parties to compute a function on their joint private inputs securely [Yao86]. Most of the MPC literature studies classical functionality over classical inputs with different notions of security, such as *full security*, *security with abort*, and *security with identifiable abort* [RBO89, MGW87, IOZ14].

Recently, secure multiparty quantum computation (MPQC) has raised research interest. Most of the works consider the fully quantum setting i.e., the functionality, including inputs and outputs, is quantum. Like in the classical setting, it is known that an honest majority is both sufficient [CGS02, BOCG$^+$06], and necessary [ABDR04] to achieve full security, which guarantees output delivery for everyone. In light of this, the study of MPQC protocols in the dishonest majority setting has focused on the weaker notion of *security with abort* [DNS12, DGJ$^+$20, BCKM21], which allows all honest parties to abort when they detect an attack. However, such a notion is vulnerable to a denial-of-service attack because an attacker can repeatedly induce aborts. For this reason, a more recent work [ACC$^+$21] has proposed an MPQC protocol with identifiable abort (MPQC-SWIA) that allows all honest parties to agree on the identity of a corrupted party in case of an abort. Regrettably, the identification mechanism of [ACC$^+$21] only allows the participants of the protocol to identify a malicious party. This is unsatisfactory in many practical scenarios because during a dispute, external observers are aware that two groups of people are in disagreement, but it is unclear which side is acting maliciously. Consider an instance where a client accuses a tech company of failing to provide a service and, therefore, refuses payment. Conversely, the company asserts that they have indeed provided the service. In such cases, it becomes vital to employ a publicly verifiable protocol to assess their integrity. This is especially important in the quantum setting, where each party may possess only one copy of their quantum input. Once the quantum inputs are ruined, it results in the irreversible loss of inputs for honest parties. Therefore, we consider a notion of security called *publicly verifiable identifiable abort* (PVIA) that allows everyone, including outside observers, to identify the malicious party. We ask:

Is it possible to construct MPQC with publicly verifiable identifiable abort (PVIA)?

In the classical setting, one can turn MPC-SWIA into MPC-PVIA almost for free. A publicly verifiable protocol can be obtained by requiring each party to broadcast their messages and proofs to outside observers. Unfortunately, this simple solution does not work in the quantum setting due to the no-cloning theorem. One may be tempted to turn to classical verification of quantum computation (CVQC) [Mah18, CLLW22] in order to achieve public verifiability. However, this approach is restricted to computation that is performed by a single quantum party with classical outputs, and it is unclear how it can be adapted to fit into the setting of MPQC. Furthermore, all existing MPQC protocols face an inherent difficulty in achieving PVIA because their sender-receiver mechanism cannot differentiate a malicious sender from a malicious receiver. To address this issue, we propose a new primitive called Auditable Quantum Authentication (AQA), which subverts the traditional sender-receiver mechanism and holds the sender accountable for his behavior.

While PVIA security can act as insurance for honest parties when a dishonest majority is present, it is desirable to have a stronger security notion, such as full security, if it turns out that the honest parties outnumber the malicious ones. An intriguing scenario involves reducing the maximal number of malicious parties allowed for security with abort while conditionally offering full security. Such a notion is called best-of-both-worlds (BoBW) security[1]. In the classical setting, [IKK$^+$11] constructs, for every threshold $t < \frac{n}{2}$, an MPC protocol that achieves security with abort against $n-1-t$ malicious parties and achieves full security tolerating t malicious parties. [Kat07] proved that these corruption thresholds are optimal. In the quantum setting, none of the existing MPQC protocols satisfy BoBW security. Therefore, we ask:

> *Is it possible to construct a single MPQC that achieves full security under an honest majority and is secure with abort under a dishonest majority?*

1.1 Our results

We answer both questions affirmatively in the preprocessing model, which features an offline setup that prepares input-independent auxiliary quantum states. Then, during the online protocol, parties only exchange classical bits. With this approach, the parties can create classical proofs that are accessible to everyone, which in turn facilitates PVIA. Moreover, combined with quantum error correction code (QECC), the setup can create quantum states that enable distributed computation and ultimately achieve best-of-both-worlds security. Finally, we show that our offline setup can be instantiated without requiring any trusted third party.

Our first result is an MPQC protocol secure with publicly verifiable identifiable abort (PVIA) under a trusted setup. Similar to existing MPQC works [DNS12, DGJ$^+$20, ACC$^+$21, BCKM21], we assume that parties have access to an ideal functionality cMPC for classical MPC (this model is known as the MPC-hybrid model). Here, the classical MPC is assumed to be PVIA-secure, and such an MPC can be based on a post-quantum oblivious transfer (OT) and a classical broadcast channel.

Theorem 1.1 (MPQC-PVIA with trusted setup, informal)**.** *There exists a multiparty quantum computation protocol secure with publicly verifiable identifiable abort supporting poly-size quantum circuits in the preprocessing MPC-hybrid model.*

To achieve Theorem 1.1, we propose and construct a new primitive called auditable quantum authentication (AQA) that allows a classical auditor to decide the integrity of a quantum message sender. Then, in our MPQC-PVIA protocol, the actions of the trusted auditor will be taken by classical MPC.

Our second result is a best-of-both-worlds (BoBW) MPQC protocol that achieves full security against t corruptions and satisfies security with abort against $n-1-t$ corruptions under a trusted setup. We call t as the BoBW threshold. Here, we assume our underlying classical MPC to be BoBW-secure with threshold t as well, which can be based on post-quantum OT for $t < \frac{n}{3}$ and additionally requires a classical broadcast channel for $\frac{n}{3} \le t < \frac{n}{2}$.

[1] There are different flavors of best-of-both-worlds security. For example, [Kat07, BLOO11] consider MPC protocols with full security against $\lfloor \frac{n-1}{2} \rfloor$ malicious parties and $(1/p)$-security with abort against $n-1$ malicious parties. The notion of $(1/p)$-security only requires an inverse polynomial error in distinguishing the real/ideal world.

Theorem 1.2 (BoBW-MPQC with trusted setup, informal)**.** *There exists a best-of-both-worlds multiparty quantum computation protocol of threshold t supporting poly-size quantum circuits for any $t < \frac{n}{2}$ in the preprocessing MPC-hybrid model.*

The key to arriving at best-of-both-worlds security is our protocol's compatibility with decentralized quantum computation using QECC. In particular, no single party in our protocol holds all the quantum information of a piece of data during the computation step, as opposed to prior security-with-abort protocols [DGJ+20, ACC+21, BCKM21].

Combining these two results, we obtain a BoBW-MPQC-PVIA protocol that achieves full security against t corruptions and satisfies PVIA security against $n - 1 - t$ corruptions under a trusted setup. The underlying classical MPC should be BoBW-PVIA-secure, which can be based on a post-quantum OT and a classical broadcast channel.

Theorem 1.3 (BoBW-MPQC-PVIA with trusted setup, informal)**.** *There exists a best-of-both-worlds multiparty quantum computation protocol secure with publicly verifiable identifiable abort of threshold t supporting poly-size quantum circuits for every $t < \frac{n}{2}$ in the preprocessing MPC-hybrid model.*

Furthermore, we can instantiate the setups, thus obtaining the above three results without needing a trusted setup.

Theorem 1.4 (BoBW-MPQC-PVIA without trusted setup, informal)**.** *Theorems 1.1,1.2,1.3 hold in the (standard) MPC-hybrid model.*

Our main technique for instantiating the setup is to leverage MPQC secure with identifiable abort (SWIA) protocols. Interestingly, the properties of both BoBW and PVIA can be preserved under our instantiation. Note that our instantiation is based on an MPQC-SWIA protocol which, contrasting with the previous result [ACC+21], only assumes classical MPC.

	Dishonest-Majority Regime	Honest-Majority Regime	Assumptions
[BOCG+06]	No Security	Full Security	cMPC
[DGJ+20] [BCKM21]	Security with Abort $\left(\begin{array}{l}\leq n-1\\ \text{corruptions}\end{array}\right)$	Security with Abort	cMPC
[ACC+21]	Identifiable Abort $\left(\begin{array}{l}\leq n-1\\ \text{corruptions}\end{array}\right)$	Identifiable Abort	cMPC+FHE
This Work $(0\leq t<\frac{n}{2})$	Publicly Verifiable Identifiable Abort $\left(\begin{array}{l}\leq n-1-t\\ \text{corruptions}\end{array}\right)$	Full Security $\left(\begin{array}{l}\leq t\\ \text{corruptions}\end{array}\right)$	cMPC

Table 1: Comparison of MPQC protocols.

2 Technical Overview

In this section, we first explain why public verifiability does not follow directly from existing works. Then, we put forth a novel primitive called Auditable Quantum Authentication

(AQA), which ensures the secure transmission of quantum outputs and the public identification of malicious identities within a protocol. Following a high-level understanding of AQA, we then incorporate the input encoding and computation steps together to realize MPQC-PVIA. At the end of the section, we discuss the difficulty of achieving best-of-both-worlds security and elucidate our approach to attaining a BoBW-MPQC protocol.

2.1 Why is MPQC-PVIA hard to achieve?

A first observation is that classical techniques for public verifiability cannot apply to their quantum counterparts. Existing methods for classical MPC-PVIA protocols are to commit to classical messages, provide zero-knowledge arguments over the commitments, and let outside observers check whether any party deviates from the protocol. There are several issues when adapting to MPQC in the fully quantum setting. If one considers classical commitments to quantum messages [Mah18], one cannot fulfill MPQC with purely quantum outputs because such classical commitment schemes always end with measurements. Instead, one may have to consider quantum commitments [GJMZ23]. However, quantum commitments are unlikely to be duplicated and broadcast to each party for verification because of the no-cloning theorem. In addition, zero-knowledge arguments for quantum computation (e.g., [BJSW16]) only apply to problems with a classical description. Those arguments cannot prove relations involving quantum commitments.

Another difficulty arises because we require the outside observers of MPQC to have only classical computational power. Although there is research on classical verification of quantum computation (CVQC), a seemingly similar task, CVQC needs to be more relaxed because it can only resolve computations with classical outputs conducted by a single quantum prover. The techniques of CVQC fail in the fully quantum setting. Moreover, CVQC already produces an inverse polynomial soundness error when extended from decisional problems [Mah18] to sampling problems [CLLW22]. Thus, there is little hope that CVQC can aid the construction of MPQC-PVIA.

One may try to upgrade MPQC-SWIA to MPQC-PVIA directly, but there is still a gap between them. The MPQC-SWIA protocol by [ACC$^+$21] is based on a Sequential Authentication primitive that outputs two suspects whenever message tampering is detected. However, it gives no information about the *exact* party that deviates from the protocol. The resulting MPQC-SWIA allows honest parties to agree on the same malicious party when protocol aborts, but an outside observer only sees two groups of people accusing each other. This outcome arises from the conventional utilization of quantum authentication codes[2], where the sender sends an authenticated state to the receiver, and the receiver is in charge of measuring the authentication checksum to validate the state. However, this kind of validation mechanism relies on the synergy of both the sender and the receiver over a single-copy state, which makes it challenging to achieve public verifiability. To address this, we subvert the old idea and creatively combine quantum authentication codes and quantum teleportation in a white-box manner.

[2] The prevalent approach in most existing works involves employing authentication codes in this manner [ACC$^+$21, BCKM21, DGJ$^+$20].

2.2 Our Solution: Auditable Quantum Authentication (AQA)

The primary goal of AQA is to establish a mechanism where the sender of an authenticated state is held responsible for his own sending action through a test performed by someone trustworthy. In a normal quantum authentication scheme, the receiver of an authenticated state runs the decoding algorithm to obtain either the original message or an authentication failure symbol. To learn the true authentication outcome, an outside observer has to trust the party who executes the decoding algorithm. This would require trust in the receiver, who might be malicious. To resolve this issue, we propose an *auditable* quantum authentication scheme that separates the authenticity check from the message decoding process. Importantly, the AQA scheme is equipped with a *classical* auditing algorithm that decides message authenticity and outputs a decoding key for the receiver to recover the message. With AQA, an outside observer can learn the authentication outcome by trusting a classical auditor who executes the auditing algorithm. Later on, we can replace the classical auditor with a publicly verifiable classical MPC (cMPC) to completely remove the need of trust.

AQA is designed to be cooperated by three parties: a sender, a receiver, and a classical auditor. We define AQA as consisting of five algorithms: Setup, Enc, Send, Audit, Recv. In the beginning, Setup prepares initial states for all the parties, and Enc produces an authenticated state σ for the sender. The sender runs Send(σ) to generate a classical proof pf showing that the quantum message has been delivered. Next, the auditor runs Audit(pf) to verify the proof and produce a decoding key dk. Afterward, the receiver can run Recv(dk) to obtain the quantum message. The security of AQA entails that Recv(dk) produces the correct quantum message (up to a negligible error) whenever Audit(pf) outputs a positive verification outcome.

Constructing a (Simplified) AQA We will start with a normal quantum authentication scheme (Gen,Enc,Dec), the Clifford code [ABOEM17] in particular. We aim to keep the encoding procedure Enc and split its decoding procedure into several parts. The decoding procedure of Clifford code applies a secret Clifford gate ${F^\dagger}^{M,T}$ to an authenticated state $\sigma^{M,T}$ and measures T register in the computational basis. An authentication failure occurs if the measurement result is not all zeros. Otherwise, the content of the M register will be the message state. To make this authentication scheme classically auditable, we consider the following alternative decoding procedure that involves 4 algorithms. We take $|M| = 1$ and $|T| = \lambda$ as an example.

Setup:

- Generate EPR pairs $\{(e_0^i, e_1^i)\}_{i\in[\lambda+1]}$ of length $\lambda+1$. Put $\{e_0^i\}_{i\in[\lambda+1]}$, $\{e_1^i\}_{i\in\{2,\cdots,\lambda+1\}}$, e_1^1 into the sending register S, the checking register C, and the receiving register R.
- Apply the secret Clifford gate $F^\dagger \leftarrow \mathscr{C}_{\lambda+1}$ to R, C.

Send:

- Sending Procedure: Teleport the authenticated state σ through the sending register S.
- Proving Procedure: Measure the checking register C in the computational basis.
- Set the classical proof as the teleportation Pauli P and the measurement result c.

Audit:

- Compute the Pauli $F^\dagger P F$ and express it as a tensor product of two Pauli gates $\hat{P}_R, \hat{P}_C$ that act on $1, \lambda$ qupits respectively.
- Report an authentication failure if $c \neq x(\hat{P}_C)$. Set the decoding key as the Pauli $\hat{P}_R$.

Recv:

- Apply $\hat{P}_R^{\dagger\,R}$ and output the state on the receiving register R.

Running these 4 algorithms in a row is equivalent to running the decoding procedure of Clifford code. This follows almost directly from quantum teleportation: If we denote $F^\dagger \sigma = (\rho^M, \tau^T)$ and if teleporting σ through S during Send yields teleportation result P, then the state in R, C would collapse from $F^\dagger(\{e_1^i\}_{i \in [\lambda+1]})$ to

$$F^\dagger P(\sigma) = (F^\dagger P F)(F^\dagger(\sigma)) = (\hat{P}_R \otimes \hat{P}_C)(\rho, \tau) = (\hat{P}_R(\rho), \hat{P}_C(\tau))$$

This shows that the measurement result of the checking register C equals $x(\hat{P}_C)$ if and only if the measurement result of τ equals all zeros. The current decoding procedure includes a classical algorithm Audit that determines the authentication outcome, so we can set our simplified AQA as these 4 algorithms plus the encoding algorithm of Clifford code.

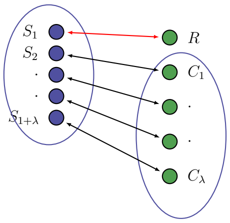


Figure 1: AQA Setup: Each edge represents an EPR pair. The nodes on the left contain halves of EPR pairs $\{e_0^i\}_{i \in [\lambda+1]}$, and the nodes on the right contain the other halves of EPR pairs $\{e_1^i\}_{i \in [\lambda+1]}$. The encircled vertices (the sending register S and checking register C) are given to the sender. The lonely vertex (receiving register R) is given to the receiver.

Proving Security of AQA For the security of AQA, it's crucial for Recv to recover the original message whenever Audit doesn't indicate an authentication failure upon receiving proof from an adversarial sender. Take, for instance, a malicious sender who alters the authenticated message σ prior to the execution of Send. In such a specific scenario, the security of the simplified AQA is derived from the established equivalence between the processes (Send, Audit, Recv) and Dec mentioned in the preceding paragraph and grounded on the efficacy of (Enc, Dec) as a quantum authentication scheme. However, the simplified AQA is presented mainly as an explanation of how we achieve the audit functionality, and is not yet secure against arbitrary malicious senders. To further protect against adversaries employing arbitrary attacks, our formal AQA in Section 5 additionally integrates quantum one-time pads. To offer a high-level intuition, the use of quantum one-time pads can split the attack of the malicious sender into a combination of Pauli attacks. Since the malicious sender knows nothing about the random Clifford key F, the Clifford twirl will transform Pauli attacks into random Pauli operators distributed across the states, breaking the consistency of P and c.

2.3 From AQA to MPQC-PVIA

From the previous section, we see that the transmission of quantum information can be audited by a classical party. We now build an MPQC-PVIA protocol with AQA where the auditing is performed by a publicly verifiable classical MPC.

MPQC-PVIA with Setup The MPQC-PVIA consists of two phases: an offline setup and an online phase. The offline setup generates EPR pairs that would allow each party to send their input to the server (who is a designated party, say P_1), and runs the setup of AQA. During the online phase, every party teleports their input to P_1 and P_1 only obtains a ciphertext of the joint inputs. Next, P_1 performs quantum computation on the ciphertext as instructed by classical MPC. Finally, P_1 sends the output ciphertexts to other parties using AQA, which is audited by classical MPC. These three steps in the online phase are called input encoding, computation, and output delivery, respectively.

We now move on to examine security. In our protocol, the parties' inputs are gathered towards P_1, and the quantum computation is solely performed by P_1. Thus, only P_1 can launch an effective attack. The attack would ruin P_1's ciphertext, and P_1 would ultimately face an authentication error when transmitting the ciphertext with AQA. In this case, the classical MPC that runs the audit algorithm can publicly output P_1 as malicious. As a result, our protocol achieves MPQC while maintaining PVIA security.

Instantiable Setup Next, we show how to instantiate our setup with an MPQC protocol secure with identifiable abort (SWIA). Note that we refine a slightly different version of MPQC-SWIA, thereby circumventing the need for the post-quantum Fully Homomorphic Encryption (FHE) assumption needed in [ACC+21].

MPQC-SWIA guarantees that whenever an abort happens, cMPC will output a partition of parties with all honest parties staying in the same group. Our approach for instantiating the setup is to run MPQC-SWIA hierarchically to prepare the states that the setup would generate. The hierarchical MPQC-SWIA maintains a grouping between parties, where all parties are initially in the same group. Each group will try to run MPQC-SWIA by themselves, and a group breaks into two whenever MPQC-SWIA fails. At some point, all parties must have succeeded in running MPQC-SWIA within their group (or they will continue running MPQC-SWIA within descent subgroups), so they can proceed to execute the online MPQC-PVIA protocol. By employing the security of MPQC-PVIA with preprocessing, it is guaranteed that either the honest parties obtain their outputs, or some malicious party in the group that contains all honest parties will be publicly identified.

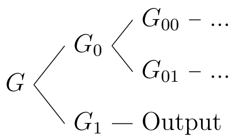


Figure 2: Hierarchical MPQC-SWIA parties try to run the offline setup using MPQC-SWIA. Initially, G contains all the parties. When the first MPQC-SWIA run by G terminates with a failure, parties in G separate into two groups G_0 and G_1, who run another MPQC-SWIA within their own group. In this figure, G_1 executes MPQC-SWIA successfully and obtains the setup output. They can proceed to execute the online MPQC-PVIA protocol.

2.4 Best-of-Both-Worlds Security

One advantage of our protocol design is its flexibility to provide best-of-both-worlds security. That is, we construct an MPQC protocol that simultaneously achieves full security when there are at most $t < \frac{n}{2}$ corruptions and satisfies security with publicly verifiable identifiable abort against at most $n - 1 - t$ corrupted parties.

Prior to this work, the honest-majority and the dishonest-majority worlds were once separated because of a tension between sharing and extracting quantum information. We elaborate on it as follows. MPQC protocols that obtain full security in an honest majority setting [BOCG+06] are based on verifiable quantum secret sharing (VQSS). In these protocols, each party individually creates VQSS of their input and distributes the shares across parties. The problem is that the secret shares sent between malicious parties are private information. Once the number of corrupted parties reaches one-half, the simulator cannot extract the adversary's input from the available secret shares. This is also why current MPQC protocols designed for a dishonest majority need every quantum message to be transmitted through all parties: the simulator can extract inputs when the quantum message passes through an honest party. As a result, protocols against a dishonest majority cannot divide a piece of quantum information across multiple parties, and a single malicious party is sufficient to destroy the information subjected to the computation.

Our solution to this tension is to utilize the offline-online structure of our protocol and incorporate quantum error correction codes (QECC). First, our offline setup[3] prepares QECC codewords on EPR pairs and distributes the codewords evenly across parties. Afterward, the parties can perform distributed computation over QECC codewords in the online protocol. In this protocol, the honestly generated QECC codewords facilitate the sharing of quantum information. Moreover, to extract quantum information even in the presence of a malicious majority, the setup can entangle a trapdoor with the states prepared for the parties and use the trapdoor to extract online inputs. We see that the offline setup acts as a vital piece of machinery that allows information extraction while preparing for the online distributed computation.

Our BoBW-MPQC protocol is reminiscent of the classical BoBW-cMPC protocols [IKK+11, Kat07, BLOO11]. A key difference is that the classical protocols need to broadcast secret sharings and invoke the ideal functionality on the inputs multiple times, both of which are

[3] Similar to the previous subsection, this setup can be instantiated using MPQC-SWIA.

infeasible in MPQC due to no-cloning. Our protocol does not follow the same pattern, and we achieve the same goal in the merit of quantum teleportation.

3 Preliminary

Let $[n] = \{1, \cdots, n\}$. We denote by $A_{[n]}$ the tuple $(A_1, \cdots, A_n)$. Uniform sampling from a set S is denoted by $s \leftarrow S$. A function $f\colon \mathbb{N} \to [0, 1]$ is called negligible, if for every polynomial $\mathrm{poly}(\cdot)$ and all sufficiently large n, it holds that $f(n) < |1/\mathrm{poly}(n)|$. We use $\mathrm{negl}(\cdot)$ to denote an unspecified negligible function.

Quantum states are written in lowercase Greek alphabets, e.g., ρ, σ. Quantum operations are written in uppercase Latin alphabets, e.g., U, V. We write ρ^{M} and U^{M} to specify that ρ is stored in register M and U operates on register M. The notation (ρ, σ) denotes a state on two registers that may be entangled. The letters QPT stands for quantum polynomial time.

Fix a prime p. A qupit in pure state $|\phi\rangle$ is a unit vector in the p-dimensional Hilbert space $\mathbb{C}^p$ and can be identified with the density operator $\mathsf{Mixed}\left[|\phi\rangle\right] := |\phi\rangle\langle\phi|$. The set of n-qupit mixed states, denoted $\mathcal{D}^n$, consists of positive semi-definite operators on $\mathbb{C}^{p^n}$ with trace 1. We sometimes identify a mixed state ρ with its purification, which is a pure state $|\phi\rangle$ such that $\mathsf{Mixed}\left[|\phi\rangle\right]$ has partial trace ρ. We also consider sub-normalized mixed states, which are positive semi-definite operators with trace at most 1. We identify a distribution $\{\rho_j\}$ of sub-normalized states with the state $\mathbb{E}_j \, \rho_j$. Two sequences of sub-normalized states $\rho(n), \sigma(n) \in \mathcal{D}^{\mathrm{poly}(n)}$ are said to be statistically indistinguishable, denoted $\rho \approx \sigma$, if they have trace distance $\mathrm{tr}\,|\rho(n) - \sigma(n)| = \mathrm{negl}(n)\,\mathrm{tr}(\rho(n))$.

3.1 Quantum Computation

A quantum operation is a completely positive, trace preserving (CPTP) map acting on mixed states. Any such map can be represented as $\{A_j\}$ which maps a mixed state ρ to the mixed state $\sum_j A_j \rho A_j^\dagger$. Each A_j defines a completely positive (CP) map $\rho \mapsto A_j \rho A_j^\dagger$. For example, the measurement operator $\{|j\rangle\langle j|\}_{j\in\mathbb{Z}_p}$ in the computation basis is a CPTP map, whereas each projector $|j\rangle\langle j|$ is only a CP map. Every unitary operator defines a CPTP map.

Consider the phase $\omega = e^{2\pi i/p}$, the shift operator $\mathsf{X} : |j\rangle \mapsto |j+1\rangle$ and the clock operator $\mathsf{Z} : |j\rangle \mapsto \omega^j |j\rangle$. Write $\mathsf{X}^{(x_1,\cdots,x_n)}\mathsf{Z}^{(z_1,\cdots,z_n)} = \bigotimes_{j\in[n]} \mathsf{X}^{x_j}\mathsf{Z}^{z_j}$ where each $x_j, z_j \in \mathbb{Z}_p$. We define the Pauli basis $\mathscr{P}_n^* = \{\mathsf{X}^x\mathsf{Z}^z \mid x, z \in \mathbb{Z}_p^n\}$, which is a basis for the space of linear operators on $\mathbb{C}^{p^n}$. Decomposing a linear operator according to this basis is called the Pauli decomposition. For convenience, we identify the Pauli $P_a = \mathsf{X}^{x_a}\mathsf{Z}^{z_a} \in \mathscr{P}_n^*$ with the string $(z_a, x_a) = (z(P_a), x(P_a)) \in \mathbb{Z}_p^{2n}$. Define the Pauli group $\mathscr{P}_n$ as $\{\omega^k\mathsf{X}^x\mathsf{Z}^z \mid k \in \mathbb{Z}_p,\ x, z \in \mathbb{Z}_p^n\}$ and the Clifford group $\mathscr{C}_n$ as the normalizer of $\mathscr{P}_n$ in the unitary group quotient by global phases. That is, a unitary $C \in \mathscr{C}_n$ if and only if for all $A \in \mathscr{P}_n$, $CAC^\dagger \in \mathscr{P}_n$. Intuitively, it means that with a reasonable update of the Pauli gate, we can swap the order where a Clifford gate and a Pauli gate are applied.

The Clifford group is generated by the Fourier transform gate $\mathsf{H} : |j\rangle \mapsto \frac{1}{\sqrt{p}}\Sigma_k\, \omega^{jk}\,|k\rangle$, the phase gate $\mathsf{S} : |j\rangle \mapsto \omega^{j(j-1)/2}\,|j\rangle$ and the sum gate $\mathsf{CX} : |j, k\rangle \mapsto |j, k+j\rangle$ [Cla06]. When $p = 2$, the phase gate is defined as $\mathsf{S} : |j\rangle \mapsto i^j\,|j\rangle$ instead. One can sample uniformly random

Clifford gates in polynomial time [VDB21, GYW23]. We will write $\mathsf{CX}_{(b_1,\cdots,b_n)}{}^{R_0,\cdots,R_n}$ as the abbreviation of $\mathsf{CX}^{b_1}{}^{R_0,R_1}\cdots\mathsf{CX}^{b_n}{}^{R_0,R_n}$.

Universal quantum computation can be carried out with Clifford gates and T gates, where $\mathsf{T} : |j\rangle \mapsto e^{\frac{2\pi i\eta_j}{p^2}}|j\rangle$ with $\eta_j = p\binom{j}{3} - j\binom{p}{3} + \binom{p+1}{4}$ [CAB12]. When $p = 2$, the T gate is defined as $\mathsf{T} : |j\rangle \mapsto e^{\frac{\pi ij}{4}}|j\rangle$ instead. Although T gate is not in the Clifford group, it can be applied using classically controlled Clifford operations with the help of the T state $|\mathsf{T}\rangle = \mathsf{T}|+\rangle$, where $|+\rangle = \frac{1}{\sqrt{p}}\sum_{j=0}^{p-1}|j\rangle$. T states can be purified from noisy ones using classically controlled Clifford gates [BK05, CAB12]. We note that the phase gate S can also be applied using classically controlled $\mathsf{X}, \mathsf{Z}, \mathsf{CX}$ gates with the help of the state $|\mathsf{S}\rangle = \mathsf{S}|+\rangle$.

3.2 Quantum One-Time Pad

Definition 3.1. A quantum one-time pad (QOTP) with key $P \in \mathscr{P}_n$ is a symmetric-key encryption scheme that consists of the following two algorithms.

- Encryption: $\mathsf{QOPT.Enc}_P(\rho) := P\rho P^\dagger$.
- Decryption: $\mathsf{QOPT.Dec}_P(\rho) := P^\dagger\rho P$.

It is well known that the ciphertext under QOTP is maximally mixed:

Lemma 3.2 (Pauli Twirl)**.** *For every state* $|\phi\rangle^{M,N} = \sum_u |u\rangle^M \otimes |\phi_u\rangle^N$ *it holds that*

$$\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[P^M|\phi\rangle^{M,N}\right] = \left(\mathop{\mathbb{E}}_{r\leftarrow\mathbb{Z}_p^n} \mathsf{Mixed}\left[|r\rangle\right]\right)^M \otimes \left(\sum_u \mathsf{Mixed}\left[|\phi_u\rangle\right]\right)^N$$

The same result holds when P is randomly sampled from the Clifford group $\mathscr{C}_n$. Moreover, it is well known that QOTP can split a quantum attack into a probabilistic combination of Pauli attacks. This work considers a specific scenario where an untrusted party measures a state which is protected under QOTP. We formulate the following lemma, which shows that any attack would be equivalent to a probabilistic combination of Pauli attacks that cause different shifts. We prove the lemma in Appendix A.

Lemma 3.3 (Pauli Twirl with Measurement)**.** *Let* $|\phi\rangle^{M,N} = \sum_{u\in\mathbb{Z}_p^n} |u\rangle^M \otimes |\phi_u\rangle^N$ *be a state and* $v \in \mathbb{Z}_p^n$ *be the target measurement result. For any attack* $A^{M,N} = \sum_{Q\in\mathscr{P}_n^*}\left(Q^M \otimes A_Q{}^N\right)$ *applied on the QOTP-protected state, it holds that*

$$\begin{aligned}&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[|v + x(P)\rangle\langle v + x(P)|^M A^{M,N} P^M |\phi\rangle^{M,N}\right]\\ &= \mathop{\mathbb{E}}_{r\leftarrow\mathbb{Z}_p^n} \sum_u \mathsf{Mixed}\left[\left(\sum_{x(Q)=v-u} Q \otimes A_Q\right)\left(|r\rangle \otimes |\phi_u\rangle\right)\right]\end{aligned}$$

3.3 Quantum Authentication Code

Quantum authentication code detects whether unauthorized alterations have been made to the data. When alternation is detected, the algorithm will output a rejection symbol $\perp$.

Definition 3.4 (Quantum Authentication Code, [BCG+02]). A quantum authentication code consists of three algorithms. The key generation algorithm Gen takes in the security parameter 1^λ and the message size 1^ℓ and outputs a random secret key sk. The encoding algorithm Enc maps a secret key sk and a quantum message on M to a quantum ciphertext on MT. The decoding algorithm Dec maps a secret key sk and a quantum ciphertext on MT to a quantum message M. These algorithms should satisfy the following properties.

- Completeness: For every secret key sk, it holds that $\mathsf{Dec}_{\mathsf{sk}} \circ \mathsf{Enc}_{\mathsf{sk}} = \mathbb{1}$.
- Security: For any quantum map $\mathcal{A}$, there exists two CP maps $\mathcal{A}_{\mathsf{Acc}}$ and $\mathcal{A}_{\mathsf{Rej}}$ such that $\mathcal{A}_{\mathsf{Acc}} + \mathcal{A}_{\mathsf{Rej}}$ is trace preserving and that for any (possibly entangled) states $\rho, \rho_{\mathsf{aux}}$,

$$\left\{ (\rho',\rho'_{\mathsf{aux}}) \;\middle|\; \begin{array}{c} \mathsf{sk}\leftarrow\mathsf{Gen}(1^\lambda,1^\ell) \\ \sigma\leftarrow\mathsf{Enc}(\mathsf{sk},\rho) \\ (\sigma',\rho'_{\mathsf{aux}})\leftarrow\mathcal{A}(\sigma,\rho_{\mathsf{aux}}) \\ \rho'\leftarrow\mathsf{Dec}(\mathsf{sk},\sigma') \end{array} \right\} \underset{\mathrm{negl}(\lambda)}{\approx} \left(\Big(\rho, \mathcal{A}_{\mathsf{Acc}}(\rho_{\mathsf{aux}})\Big) + \Big(|\bot\rangle\langle\bot|, \mathcal{A}_{\mathsf{Rej}}(\rho_{\mathsf{aux}})\Big) \right)$$

Here, we recall the Clifford authentication code from [ABOEM17]. The key generation algorithm outputs a uniformly random Clifford gate $E^{M,T}$. The encoding procedure augments the message state ρ^M with traps $|0\rangle^{\otimes\lambda\,T}$ and applies $E^{M,T}$. The decoding procedure applies $E^\dagger$ followed by measuring the register T in the computational basis. If the measurement results are not all zero, the content of M is replaced with $|\bot\rangle$. The Clifford authentication code satisfies Definition 3.4. The following lemma is crucial to its proof, and we will use the lemma directly later on.

Lemma 3.5 (Pauli Partitioning by Clifford, [ABOEM17, Cha05]). *For every Pauli operators $Q, Q' \in \mathscr{P}_n$ that do not lie in $\{\omega^k I \mid k \in \mathbb{Z}_p\}$, it holds that*

$$\Pr_{C\leftarrow\mathscr{C}_n}\left[C^\dagger Q C = Q'\right] = \mathrm{negl}(n).$$

The Clifford code also supports homomorphic computation for any Clifford operation. Consider a Clifford-code ciphertext $\mathsf{Enc}_E(\rho^M)$ with secret key $\mathsf{sk} = E$. To perform a Clifford gate G on ρ, it suffices to update the secret key as $\mathsf{sk}' = EG^\dagger$. This works because we have $\mathsf{Enc}_{\mathsf{sk}}(\rho) = E(\rho, |0\rangle^{\otimes\lambda}) = EG^\dagger(G\rho, |0\rangle^{\otimes\lambda}) = \mathsf{Enc}_{\mathsf{sk}'}(G(\rho))$.

3.4 Quantum Error-Correction Code

Quantum error correction code can protect quantum states from errors as long as the number of errors is limited. In this work, it suffices to consider erasure errors.

Definition 3.6 (Quantum Error Correction Code). A $[[n,k]]_p$ quantum error correction code consists of two algorithms. The encoding algorithm $\mathsf{QECC.Enc} : \mathcal{D}^k \to \mathcal{D}^n$ encodes a k-qupit message into a n-qupit codeword. The decoding algorithm $\mathsf{QECC.Dec} : \mathcal{D}^n \times \{0,1\}^n \to \mathcal{D}^k$ takes a modified codeword and its location of errors and outputs a k-qupit message. A quantum error correction code is said to correct t erasure errors, if for any $\rho \in \mathcal{D}^k$ and any quantum channel Δ^R acting on $|R| < t$ qupits, it holds that

$$\mathsf{QECC.Dec}\left(\Delta^R \mathsf{QECC.Enc}(\rho), \mathbb{1}_R\right) = \rho$$

where $\mathbb{1}_R$ specifies the locations of R among the n qupits.

To arrive at best-of-both-worlds security for any threshold $t < \frac{n}{2}$, we can use the quantum polynomial code of [ABO97], which also satisfies other desirable properties.

Lemma 3.7 (Polynomial Code, [ABO97])**.** *For every $t < \frac{n}{2}$ and prime $p > n$, there exists a $[[n,1]]_p$ quantum error correction code that corrects t erasure errors with the following additional properties:*

- *Syntax: The encoding algorithm applies a Clifford gate to input ρ and ancilla $|0\rangle^{\otimes(n-1)}$. We will denote the Clifford gate for encoding as* QECC.
- *Transversal Measurement: The decoding algorithm commutes with qupit-wise measurement in the computational basis.*
- *Fault-Tolerant Computation:* $\mathsf{X}, \mathsf{Z}, \mathsf{CX}, \mathsf{H}$ *gates and their inverses can be applied to the underlying message ρ by locally applying some of these gates (and measurements) to the individual components of the codeword (using ancillas).*

A state injection technique (Appendix B) shows that the S gate can be performed through $\mathsf{X}, \mathsf{Z}, \mathsf{CX}^{-1}$ gates and measurements in the computational basis using ancillas. Combining with the transversal measurement and fault-tolerant computation stated above, the entire Clifford group $\mathscr{C}_n = \langle \mathsf{S}, \mathsf{H}, \mathsf{CX}\rangle$ can be applied fault-tolerantly using ancillas under the polynomial code of Lemma 3.7.

3.5 Quantum Teleportation

Quantum teleportation allows parties to transmit quantum messages using only classical communication and pre-shared quantum states. Below, an EPR pair $({e_S}^{S_1,\cdots,S_n}, {e_R}^{R_1,\cdots,R_n})$ of length n stands for the state $\bigotimes_{j\in[n]} |\Phi^+\rangle^{S_j,R_j}$ where $|\Phi^+\rangle = \frac{1}{\sqrt{p}}\Sigma_{j=0}^{p-1} |j,j\rangle$.

Definition 3.8 (Quantum Teleportation Without Measurement)**.** Let (e_S, e_R) be an EPR pair of length n pre-shared between a sender holding input $\psi \in \mathcal{D}^n$ and a receiver. Quantum teleportation consists of two algorithms. We will also abbreviate $\mathsf{TP.Send}$ as TP.

- $\mathsf{TP.Send}(\psi^M, {e_S}^S)$ applies $\mathsf{H}^M\mathsf{CX}^{\dagger M,S}$ to $(\psi^M, {e_S}^S)$ and outputs registers M, S.
- $\mathsf{TP.Recv}(z, x, {e_R}^R)$ applies $(\mathsf{X}^x\mathsf{Z}^z)^\dagger$ to ${e_R}^R$ and outputs register R.

When we speak of teleporting a state ψ^M via register S, we mean to apply $\mathsf{TP.Send}^{M,S}$, measure (M, S) in the computational basis and interpret the measurement result (z, x) as the Pauli $\mathsf{X}^x\mathsf{Z}^z$. The following lemma states that the teleportation result $\mathsf{X}^x\mathsf{Z}^z$ can help the receiver recover the original quantum message ψ. A proof can be found in Appendix A.

Lemma 3.9. *Let (ψ, τ) be a purified state independent of (e_S, e_R). Then*

$$\big(\mathsf{TP.Send}(\psi^M, {e_S}^S), {e_R}^R, \tau^N\big) = \frac{1}{p^n} \sum_{x,z\in\mathbb{Z}_p^n} |z\rangle^M \otimes |x\rangle^S \otimes \Big((\mathsf{X}^x\mathsf{Z}^z)\,\psi^R, \tau^N\Big)$$

4 Model and Definition

We focus on interactive protocols between n parties $\mathsf{P}_1, \cdots, \mathsf{P}_n$ with quantum computational power. They can communicate using pairwise authenticated quantum channels and a broadcast channel for classical messages. We work in the synchronous communication model where the protocol proceeds in rounds, and each message will certainly arrive at the end of each round. In addition, we consider the presence of a protocol observer O who passively receives and records classical information from the broadcast channel all the time. The adversary $\mathcal{A}$ can statically corrupt a set $I \subset \{\mathsf{P}_1, \cdots, \mathsf{P}_n\}$ of up to $n-1$ parties.

The quantum computation to be performed is modeled as a quantum circuit C, which takes n parts of quantum inputs and produces n parts of quantum outputs. Without loss of generality, we assume that the corresponding inputs and outputs have equal size. We always apply [BK05, CAB12] to convert C into the following format, incurring only a polynomial growth in description size. The ancilla ϕ_{anc} consists of $|0\rangle, |\mathsf{S}\rangle, |\mathsf{T}\rangle$ states, and the circuit operates on a total of $\ell_{\text{total}} = \sum_i \ell_i + d$ qupits.

Specification of Quantum Circuit C

1. Take input registers $R_1, \cdots, R_n$ where $|R_i| = \ell_i$.
2. Initialize register $N = (N_1, \cdots, N_d)$ as some ancilla (magic) state ϕ_{anc} where $|N_i| = 1$.
3. For $k = d, \cdots, 1$, perform the following computation, denoted as $C[k]$:
   (a) Measure N_k in the computational basis and obtain a pit b_k.
   (b) Compute a classical circuit $f_k(b_k, \cdots, b_d)$ that outputs a Clifford gate G_{k-1}.
   (c) Apply $G_{k-1} \in \mathscr{C}_{\ell_1+\cdots+\ell_n+k-1}$ on registers $(R_1, \cdots, R_n, N_1, \cdots, N_{k-1})$.
4. Output registers $R_1, \cdots, R_n$.

We will also consider evaluating a quantum circuit in a fault-tolerant manner. Under the quantum error-correction code of Lemma 3.7, the Clifford group can be applied fault-tolerantly with ancillas. Hence, a quantum circuit can be written in the above form, with each G_{k-1} additionally admits a fault-tolerant expression $(G_{k-1}^{(1)}, \cdots, G_{k-1}^{(n)})$ consisting of Clifford gates such that

$$(G_{k-1}^{(1)} \otimes \cdots \otimes G_{k-1}^{(n)}) \circ \mathsf{QECC.Enc} = \mathsf{QECC.Enc} \circ G_{k-1}$$

4.1 The Ideal World of BoBW-MPQC-PVIA

A multi-party quantum computation protocol is defined using the real vs. ideal paradigm. In the ideal world, the parties delegate the computation C to a trusted party T. The only way for the corrupted parties to interrupt the delegation is to ask T to abort, in which case T publicly announces their identities. The ideal world of best-of-both-worlds multi-party quantum computation secure with publicly verifiable identifiable abort is formally defined as follows. We denote its joint output distribution as $\mathsf{Ideal}^{\mathsf{MPQC}}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\lambda, t, C, \rho_1, \cdots, \rho_n)$.

$\mathsf{Ideal}^{\mathsf{MPQC}}$: **Best-of-Both-Worlds Multi-party Quantum Computation Secure with Publicly Verifiable Identifiable Abort**

Common input:

The security parameter 1^λ, fault-tolerance threshold t and quantum circuit C.

Input:

P_i holds input ρ_i. $\mathcal{A}_I$ holds input ρ_{aux} and controls parties in I.

T receives inputs and performs computation:

Each party P_i sends some $\tilde{\rho}_i$ as input to T. Honest parties choose $\tilde{\rho}_i = \rho_i$.
T computes $(\rho'_1, \cdots, \rho'_n) \leftarrow C(\tilde{\rho}_1, \cdots, \tilde{\rho}_n)$.

T sends back outputs:

T sends ρ'_i to all $\mathsf{P}_i \in I$.
$\mathsf{P}_i \in I$ can send `abort` message to T. Let J be the set of parties who indeed do so.
If $|J| > t$, T publicly aborts to J.
If $|J| \leq t$, T sends ρ'_i to all $\mathsf{P}_i \notin I$.

Output:

Honest parties output whatever output received from T.
The observer O outputs whatever public information received from T.
The adversary $\mathcal{A}_I$ outputs a function of his view.

For a protocol Π, we denote by $\mathsf{Real}^{\Pi}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\lambda, \rho_1, \cdots, \rho_n)$ the joint output distribution of the honest parties, the observer, and the adversary at the end of protocol Π when executed by $\mathsf{P}_i(\rho_i)$ in the presence of an adversary $\mathcal{A}_I(\rho_{\mathsf{aux}})$ corrupting parties in I. For a protocol Π with a trusted setup Σ, we define $\mathsf{Real}^{\Pi\circ\Sigma}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\lambda, \rho_1, \cdots, \rho_n)$ similarly with Σ being executed by a trusted party prior to Π.

Definition 4.1. We say that a protocol Π is a best-of-both-worlds multi-party quantum computation secure with publicly verifiable identifiable abort (BoBW-MPQC-PVIA) of threshold t over a circuit C, if for every $|I| \leq n - 1 - t$ and every non-uniform (QPT) adversary $\mathcal{A}_I$ corrupting parties in I, there is a non-uniform (QPT) simulator Sim_I corrupting parties in I, such that for any quantum inputs $\rho_i \in \mathcal{D}^{\ell_i}$, $i \in [n]$,

$$\mathsf{Real}^{\Pi}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\lambda, \rho_1, \cdots, \rho_n) \approx \mathsf{Ideal}^{\mathsf{MPQC}}_{\mathsf{Sim}_I(\rho_{\mathsf{aux}})}(1^\lambda, t, C, \rho_1, \cdots, \rho_n)$$

If the protocol Π has a trusted setup Σ, the indistinguishability requirement is replaced with

$$\mathsf{Real}^{\Pi\circ\Sigma}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\lambda, \rho_1, \cdots, \rho_n) \approx \mathsf{Ideal}^{\mathsf{MPQC}}_{\mathsf{Sim}_I(\rho_{\mathsf{aux}})}(1^\lambda, t, C, \rho_1, \cdots, \rho_n)$$

Definition 4.2. We say that Π is a multi-party quantum computation secure with publicly verifiable identifiable abort (MPQC-PVIA) over a circuit C if Definition 4.1 holds for $t = 0$.

4.2 (Preprocessing) MPC-Hybrid Model

Following [DNS12, DGJ+20, ACC+21, BCKM21], we assume an ideal functionality cMPC for *reactive*[4] classical multiparty computation within our MPQC protocol. The ideal world of classical MPC is similar to that of MPQC defined in section Section 4.1, but allows only classical messages and classical computation. In additional to producing n private outputs, the classical computation is allowed to generate an additional output which the ideal functionality publicly outputs if there is no abort. In our presentation, we will simply view cMPC as a trusted classical party. We refer this setting as the MPC-hybrid model. The preprocessing MPC-hybrid model extends the MPC-hybrid model by allowing an input-independent trusted setup to be executed prior to the actual protocol.

One can instantiate the MPC ideal functionality using a post-quantum best-of-both-worlds MPC protocol secure with publicly verifiable identifiable abort and publicly verifiable output. A concrete construction of such a protocol is to plug the MPC protocol of [BOSSV20] into the compiler of [IKK+11] to attain best-of-both-worlds security, and then apply [Unr10]'s lifting theorem to obtain post-quantum security.

5 Auditable Quantum Authentication (AQA)

This section presents a new primitive called *Auditable Quantum Authentication* (AQA) that lets a sender send quantum messages to a receiver and be accountable for his sending action. AQA is designed to identify the malicious sender only, while the receiving behavior is automatically guaranteed by successfully passing the test. In contrast to traditional quantum authentication codes like the Clifford or Trap codes, which necessitate the receiver to verify the checking bits, AQA adopts a different approach. It obliges the sender to generate the proof that should be verified by the auditor.

Definition 5.1 (Auditable Quantum Authentication)**.** An auditable quantum authentication scheme consists of the following five algorithms:

- $\mathsf{Setup}(1^\lambda, 1^\ell) \to (\mathsf{sk}, \phi_S, \phi_R)$ takes as input the security parameter λ and the message length ℓ and outputs a classical secret key sk, a quantum sending state ϕ_S and a quantum receiving state ϕ_R.
- $\mathsf{Enc}(\mathsf{sk}, \rho) \to \sigma$ takes as input a classical secret key sk, a quantum message state $\rho \in \mathcal{D}^\ell$ and outputs a quantum authenticated state σ.
- $\mathsf{Send}(\sigma, \phi_S) \to \mathsf{pf}$ takes as input a quantum authenticated state σ, a quantum sending state ϕ_S and outputs a classical proof pf.
- $\mathsf{Audit}(\mathsf{sk}, \mathsf{pf}) \to \mathsf{dk}$ is a *classical* algorithm that takes as input a secret key sk, a proof pf and outputs a decryption key dk. When the proof is invalid, dk will be set as $\perp$.
- $\mathsf{Recv}(\mathsf{dk}, \phi_R) \to \rho'$ takes as input a classical decryption key dk, a quantum receiving state ϕ_R and outputs a quantum message state ρ'. When $\mathsf{dk} = \perp$, ρ' will be set as $\perp$.

These algorithms should satisfy the following properties:

[4] It can be equipped with an internal state that may be taken into account when it is called next time.

- Sender completeness: For every quantum message state $\rho \in \mathcal{D}^\ell$, it holds that

$$\Pr\left[\mathsf{dk}{\neq}\bot \;\middle|\; \begin{array}{c}(\mathsf{sk},\phi_S,\phi_R)\leftarrow\mathsf{Setup}(1^\lambda,1^\ell)\\ \sigma\leftarrow\mathsf{Enc}(\mathsf{sk},\rho)\\ \mathsf{pf}\leftarrow\mathsf{Send}(\sigma,\phi_S)\\ \mathsf{dk}\leftarrow\mathsf{Audit}(\mathsf{sk},\mathsf{pf})\end{array}\right] = 1$$

- Receiver security: There exists algorithms $\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Audit}}$ such that $\mathsf{Recv} \circ \widetilde{\mathsf{Enc}} = \mathbb{1}$ and that for every completely positive map $\mathcal{A}$ and every possibly entangled quantum states $\rho, \rho_{\mathsf{aux}}$, it holds that

$$\left\{(\mathsf{dk},\phi_A) \;\middle|\; \begin{array}{c}(\mathsf{sk},\phi_S,\phi_R)\leftarrow\mathsf{Setup}(1^\lambda,1^\ell)\\ \sigma\leftarrow\mathsf{Enc}(\mathsf{sk},\rho)\\ (\mathsf{pf},\phi_A)\leftarrow\mathcal{A}(\sigma,\phi_S,\phi_R,\rho_{\mathsf{aux}})\\ \mathsf{dk}\leftarrow\mathsf{Audit}(\mathsf{sk},\mathsf{pf})\end{array}\right\} \underset{\mathrm{negl}(\lambda)}{\approx} \left\{(\mathsf{dk},\phi_A) \;\middle|\; \begin{array}{c}(\mathsf{sk},\phi_S,\sigma)\leftarrow\widetilde{\mathsf{Setup}}(1^\lambda,1^\ell)\\ (\mathsf{dk},\phi_R)\leftarrow\widetilde{\mathsf{Enc}}(\rho)\\ (\mathsf{pf},\phi_A)\leftarrow\mathcal{A}(\sigma,\phi_S,\phi_R,\rho_{\mathsf{aux}})\\ \mathsf{dk}\leftarrow\bot \text{ if } \bot\leftarrow\widetilde{\mathsf{Audit}}(\mathsf{sk},\mathsf{pf})\end{array}\right\}$$

Sender completeness guarantees that the honest sender always passes the audit. Receiver security is defined through the indistinguishability of two kinds of executions, implying that whatever property is satisfied by the right-hand side will also hold up to a negligible error for the left-hand side.

In particular, receiver security captures the following properties. First, it guarantees that the sender's inputs (σ, ϕ_S) are as if they can be generated independently of ρ, and hence contain no information about the message ρ. Similarly, the receiver's input ϕ_R is as if it already encodes ρ. Second, dk is the only information required for the receiver to recover ρ from ϕ_R, and the honest receiver always obtains the true message given that the audit accepts. Third, all the adversarial sender can do, even if the sender and the receiver collude, is to completely destroy dk at the cost of making the audit output $\bot$ at the same time.

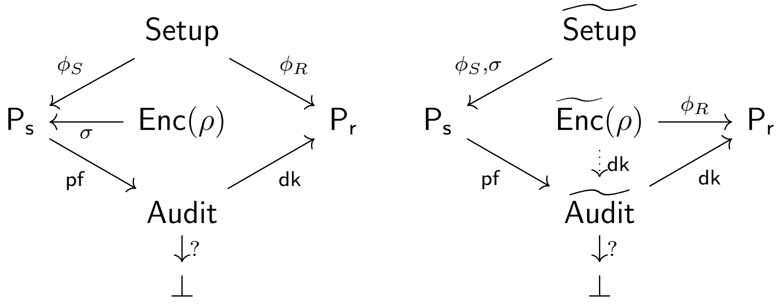


Figure 3: AQA real world (left) and ideal world (right)

5.1 Construction

Construction 1 Clifford-Form AQA

- $\mathsf{Setup}(1^\lambda, 1^\ell)$:

1. Sample random Clifford $F \leftarrow \mathscr{C}_{\ell+\lambda}$ and Pauli $P_M, P_S \leftarrow \mathscr{P}_{\ell+\lambda}, P_R \leftarrow \mathscr{P}_\ell, P_C \leftarrow \mathscr{P}_\lambda$.
2. Prepare EPR pairs on registers $(\hat{S}, (\hat{R}, \hat{C}))$ with $|\hat{S}| = \ell + \lambda$, $|\hat{R}| = \ell$, $|\hat{C}| = \lambda$.
3. Apply ${P_S}^{\hat{S}} {P_R}^{\hat{R}} {P_C}^{\hat{C}} {F^\dagger}^{\hat{R},\hat{C}}$. Name the resulting state as ${\phi_S}^{\hat{S},\hat{C}}$ and ${\phi_R}^{\hat{R}}$.
4. Output $(\mathsf{sk}, \phi_S, \phi_R)$ where $\mathsf{sk} = (F, P_M, P_S, P_C, P_R)$.

- $\mathsf{Enc}(\mathsf{sk}, \rho)$: Parse $\mathsf{sk} = (F, P_M, P_S, P_C, P_R)$ and output $\sigma = P_M F(\rho \otimes 0^\lambda) F^\dagger P_M^\dagger$.
- $\mathsf{Send}(\sigma^{\hat{M}}, {\phi_S}^{\hat{S},\hat{C}})$:
  1. Teleport $\sigma^{\hat{M}}$ via $\hat{S}$ and obtain the teleportation result $\hat{P}$.
  2. Measure $\hat{C}$ in the computational basis and obtain the measurement result $\hat{c}$.
  3. Output $\mathsf{pf} = (\hat{P}, \hat{c})$.
- $\mathsf{Audit}(\mathsf{sk}, \mathsf{pf})$:
  1. Parse $\mathsf{sk} = (F, P_M, P_S, P_C, P_R)$ and $\mathsf{pf} = (\hat{P}, \hat{c})$.
  2. Compute the quantum one-time pad $P_{M,S} := \mathsf{TP}(P_M \otimes P_S)\mathsf{TP}^\dagger$. Set the decoded teleportation result $\hat{P}'$ as the Pauli with string representation $\big(z(\hat{P}), x(\hat{P})\big) \oplus x(P_{M,S})$.
  3. Split the twirled Pauli $F^\dagger \hat{P}' F \in \mathscr{P}_{\ell+\lambda}$ as $\hat{P}'_R, \hat{P}'_C$ that act on ℓ, λ qupits respectively.
  4. If $\hat{c} \neq x(\hat{P}'_C) \oplus x(P_C)$, output $\mathsf{dk} = \bot$. Otherwise, output $\mathsf{dk} = P_R \hat{P}'_R$.
- $\mathsf{Recv}(\mathsf{dk}, \phi_R)$: If $\mathsf{dk} \neq \bot$, parse dk as a Pauli gate and output $\rho' = \mathsf{dk}^\dagger(\phi_R)\mathsf{dk}$.

In light of the homomorphic property of the Clifford code (Section 3.3), we additionally define EncG which extracts the encoding Clifford gate from the secret key. It will become useful in applications that make use of homomorphic computation.

- $\mathsf{EncG}(\mathsf{sk})$: Parse $\mathsf{sk} = (F, P_M, P_S, P_C, P_R)$ and output $P_M F$.

5.2 Security

Theorem 5.2. Construction 1 *is an Auditable Quantum Authentication scheme.*

Proof. We take the following steps. First, we analyze the state (dk, ϕ_A) that results from executing $(\mathsf{Setup}, \mathsf{Enc}, \mathcal{A}, \mathsf{Audit})$ in a row. Second, we prove sender completeness by plugging in $\mathcal{A} = \mathsf{Send}$. Third, we show that the distribution of (dk, ϕ_A) generated above is indistinguishable from a simpler state. Last, we construct $\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Audit}}$ that satisfy the requirements of receiver security.

Step 1. Without loss of generality, we assume that $\mathcal{A}$ has the same output length in ϕ_A as the input length. We can also assume that $\mathcal{A}(\sigma^{\hat{M}}, {\phi_S}^{\hat{S},\hat{C}}, {\phi_R}^{\hat{R}}, {\rho_{\mathsf{aux}}}^{\hat{W}})$ produces the classical proof pf by measuring the registers $(\hat{M}, \hat{S}, \hat{C})$ in the computational basis. Moreover, we can assume that the CP map $\mathcal{A}$ takes the form $\tau \mapsto A \tau A^\dagger$ because every CP map can be decomposed into a sum of such operators and indistinguishability of subnormalized states extends under addition. We denote the Pauli decomposition of $A' = A \ \mathsf{TP}^{\dagger \hat{M}, \hat{S}}$ as $\sum_{Q \in \mathscr{P}^*} Q^{\hat{M}, \hat{S}, \hat{C}} \otimes {A'_Q}^{\hat{R}, \hat{W}}$.

Let $(\hat{\mathcal{S}}, (\hat{\mathcal{R}}, \hat{\mathcal{C}}))$ be the EPR pair prepared during Setup. By defintion, the execution of Setup and Enc yields a random classical key $\mathsf{sk} = (F, P_M, P_S, P_C, P_R)$ and quantum state

$$\left(\sigma^{\hat{M}}, {\phi_S}^{\hat{S}, \hat{C}}, {\phi_R}^{\hat{R}}\right) = \ \mathsf{Mixed}\left[\left(P_M F \left(\rho, |0\rangle^{\otimes \lambda}\right)^{\hat{M}}, P_S \hat{\mathcal{S}}^{\hat{S}}, (P_R \otimes P_C)\, F^\dagger \left(\hat{\mathcal{R}}, \hat{\mathcal{C}}\right)^{\hat{R}, \hat{C}}\right)\right]$$

After the execution of $\mathcal{A}$, the joint state is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{sk}} \ \mathsf{Mixed}\left[A\left(P_M F \left(\rho, |0\rangle^{\otimes \lambda}\right)^{\hat{M}}, P_S \hat{\mathcal{S}}^{\hat{S}}, (P_R \otimes P_C)\, F^\dagger \left(\hat{\mathcal{R}}, \hat{\mathcal{C}}\right)^{\hat{R}, \hat{C}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right) \otimes |\mathsf{sk}\rangle\right] \\
=&\mathop{\mathbb{E}}_{\mathsf{sk}} \ \mathsf{Mixed}\left[A'\left(\mathsf{TP}\left(P_M F \left(\rho, |0\rangle^{\otimes \lambda}\right)^{\hat{M}}, P_S \hat{\mathcal{S}}^{\hat{S}}\right), (P_R \otimes P_C)\, F^\dagger \left(\hat{\mathcal{R}}, \hat{\mathcal{C}}\right)^{\hat{R}, \hat{C}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right) \otimes |\mathsf{sk}\rangle\right] \\
=&\mathop{\mathbb{E}}_{\mathsf{sk}} \ \mathsf{Mixed}\left[A'\left(P_{M,S} \mathsf{TP}\left(F \left(\rho, |0\rangle^{\otimes \lambda}\right)^{\hat{M}}, \hat{\mathcal{S}}^{\hat{S}}\right), (P_R \otimes P_C)\, F^\dagger \left(\hat{\mathcal{R}}, \hat{\mathcal{C}}\right)^{\hat{R}, \hat{C}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right) \otimes |\mathsf{sk}\rangle\right] \\
=&\mathop{\mathbb{E}}_{\mathsf{sk}} \ \mathsf{Mixed}\left[A'\left(\sum_P \frac{1}{p^{\ell+\lambda}} \left(P_{M,S} \, |z(P), x(P)\rangle^{\hat{M}, \hat{S}}, (P_R \otimes P_C)\, F^\dagger P F \left(\rho, |0\rangle^{\otimes \lambda}\right)^{\hat{R}, \hat{C}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right)\right) \otimes |\mathsf{sk}\rangle\right]
\end{aligned}$$

The first and second equalities follows from the definitions of A' and $P_{M,S}$, and the last equality is by quantum teleportation (lemma 3.9). Let us define the linear functions

$$\begin{aligned}
L_F(P) &:= \left(z(P), x(P), x(F^\dagger P F)_{[\ell+1:\ell+\lambda]}\right) \in \mathbb{Z}_p^{2\ell+3\lambda} \\
K_F(P) &:= \left(F^\dagger P F\right)_{[1:\ell]} \in \mathscr{P}_\ell
\end{aligned}$$

for every Clifford operator F. The joint state can be simplified as

$$\mathop{\mathbb{E}}_{\mathsf{sk}} \ \mathsf{Mixed}\left[A'\left(\sum_P \frac{1}{p^{\ell+\lambda}} \left((P_{M,S} \otimes P_C)\, |L_F(P)\rangle^{\hat{M}, \hat{S}, \hat{C}}, P_R K_F(P) \left(\rho\right)^{\hat{R}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right)\right) \otimes |\mathsf{sk}\rangle\right]$$

The next step is to apply Audit, which checks whether the value stored in $(\hat{M}, \hat{S}, \hat{C})$ is equal to $L_F(\hat{P}) \oplus x(P_{M,S} \otimes P_C)$ for some $\hat{P} \in \mathscr{P}_{\ell+\lambda}$. If there is such a $\hat{P}$, then Audit outputs $\mathsf{dk} = P_R K_F(\hat{P})$; otherwise, it outputs $\mathsf{dk} = \bot$. To analyze the resulting state post-selected on finding $\hat{P}$, we can apply Pauli twirl with target measurement result $L_F(\hat{P})$ (lemma 3.3) using the Pauli decomposition of A'. We obtain

$$\begin{aligned}
&\mathop{\mathbb{E}}_{r,F,P_R} \sum_{P} \mathsf{Mixed}\left[\sum_{x(Q)=L_F(\hat{P}/P)} \frac{1}{p^{\ell+\lambda}} Q\,|r\rangle \otimes A'_Q\Big(P_R K_F(P)\,(\rho)\,,\rho_{\mathsf{aux}}\Big) \otimes |P_R K_F(\hat{P})\rangle\right] \\
\overset{P_R \leftarrow P_R K_F(P)}{=} &\mathop{\mathbb{E}}_{r,F,P_R} \sum_{P} \mathsf{Mixed}\left[\sum_{x(Q)=L_F(\hat{P}/P)} \frac{1}{p^{\ell+\lambda}} Q\,|r\rangle \otimes A'_Q\Big(P_R\,(\rho)\,,\rho_{\mathsf{aux}}\Big) \otimes |P_R K_F(\hat{P}/P)\rangle\right] \\
\overset{P \leftarrow \hat{P}/P}{=} &\frac{1}{p^{2(\ell+\lambda)}} \mathop{\mathbb{E}}_{r,F,P_R} \sum_{P} \mathsf{Mixed}\left[\sum_{x(Q)=L_F(P)} Q\,|r\rangle \otimes A'_Q\Big(P_R\,(\rho)\,,\rho_{\mathsf{aux}}\Big) \otimes |P_R K_F(P)\rangle\right]
\end{aligned}$$

Summing the post-selected states corresponding to every $\hat{P} \in \mathscr{P}_{\ell+\lambda}$, we obtain the state conditioned that Audit accepts.

$$\mathop{\mathbb{E}}_{r,F,P_R} \sum_{P} \mathsf{Mixed}\left[\sum_{x(Q)=L_F(P)} Q|r\rangle^{\hat{M},\hat{S},\hat{C}} \otimes A'_Q\Big(P_R\,(\rho)^{\hat{R}}, \rho_{\mathsf{aux}}{}^{\hat{W}}\Big) \otimes |P_R K_F(P)\rangle\right] \tag{1}$$

Similarly, the state conditioned that Audit rejects is

$$\mathop{\mathbb{E}}_{r,F,P_R} \sum_{x \notin \textsc{Range}(L_F)} \mathsf{Mixed}\left[\sum_{x(Q)=x} Q|r\rangle^{\hat{M},\hat{S},\hat{C}} \otimes A'_Q\Big(P_R\,(\rho)^{\hat{R}}, \rho_{\mathsf{aux}}{}^{\hat{W}}\Big) \otimes |\bot\rangle\right] \tag{2}$$

Step 2. To see sender completeness, we take $\mathcal{A} = \mathsf{Send}$, which applies $\mathsf{TP}^{\hat{M},\hat{S}}$ followed by measuring $(\hat{M}, \hat{S}, \hat{C})$ in the computational basis. The induced $A' = A\ \mathsf{TP}^\dagger$ is a measurement in the computational basis, and the Pauli decomposition of A' involves only terms with $x(Q) = 0 \in \textsc{Range}(L_F)$. For such A', expression (2) shows that Audit never outputs $\bot$.

Step 3. We claim that (1) + (2) is statistically indistinguishable to (3) + (4).

$$\mathop{\mathbb{E}}_{r,P_R} \mathsf{Mixed}\left[\sum_{x(Q)=0} Q|r\rangle^{\hat{M},\hat{S},\hat{C}} \otimes A'_Q\Big(P_R\,(\rho)^{\hat{R}}, \rho_{\mathsf{aux}}{}^{\hat{W}}\Big) \otimes |P_R\rangle\right] \tag{3}$$

$$\mathop{\mathbb{E}}_{r,P_R} \sum_{x \neq 0} \mathsf{Mixed}\left[\sum_{x(Q)=x} Q|r\rangle^{\hat{M},\hat{S},\hat{C}} \otimes A'_Q\Big(P_R\,(\rho)^{\hat{R}}, \rho_{\mathsf{aux}}{}^{\hat{W}}\Big) \otimes |\bot\rangle\right] \tag{4}$$

The trace distance between (1) and (3) is

$$\begin{aligned}
&\operatorname{tr}\left|\mathop{\mathbb{E}}_{F,r,P_R}\sum_{P\neq I}\mathsf{Mixed}\left[\sum_{x(Q)=L_F(P)}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\otimes\left|P_RK_F(P)\right\rangle\right]\right|\\
&=\operatorname{tr}\left|\sum_{x\neq 0}\sum_{P\neq I}\mathop{\mathbb{E}}_{F,r,P_R}1_{L_F(P)=x}\ \mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\otimes\left|P_RK_F(P)\right\rangle\right]\right|\\
&\leq\sum_{x\neq 0}\sum_{P\neq I}\mathop{\mathbb{E}}_{F,r,P_R}1_{L_F(P)=x}\ \operatorname{tr}\left|\ \mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\otimes\left|P_RK_F(P)\right\rangle\right]\right|\\
&=\sum_{x\neq 0}\mathop{\mathbb{E}}_{r,P_R}\left(\sum_{P\neq I}\Pr_F\left[L_F(P)=x\right]\right)\operatorname{tr}\left(\mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\right]\right)\\
&=\sum_{x\neq 0}\mathop{\mathbb{E}}_{r,P_R}\Pr\left[x\in\textsc{Range}(L_F)\right]\ \operatorname{tr}\left(\mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\right]\right)
\end{aligned}\tag{5}$$

where the third line is from triangle inequality, the fourth line is because $\mathsf{Mixed}\left[\cdot\right]$ is positive, and the last line follows from the observation that each x has at most one P such that $L_F(P)=x$. Similarly, the trace distance between (2) and (4) is

$$\mathop{\mathbb{E}}_{r,P_R}\sum_{x\neq 0}\left(1-\Pr[x\notin\textsc{Range}(L_F)]\right)\operatorname{tr}\left(\mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\right]\right)$$

which is equal to (5). By triangle inequality, the trace distance between (1)+(2) and (3)+(4) is upper bounded by two times (5), which is

$$\begin{aligned}
&\sum_{x\neq 0}\mathop{\mathbb{E}}_{r,P_R}2\Pr\left[x\in\textsc{Range}(L_F)\right]\ \operatorname{tr}\left(\mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\right]\right)\\
&\leq\operatorname{negl}(\lambda)\ \sum_{x\neq 0}\mathop{\mathbb{E}}_{s,P_R}\operatorname{tr}\left(\mathsf{Mixed}\left[\sum_{x(Q)=x}Q\left|r\right\rangle\otimes A'_Q\left(P_R(\rho),\rho_{\mathsf{aux}}\right)\right]\right)\\
&=\operatorname{negl}(\lambda)\ \operatorname{tr}\left(4\right)\ \leq\ \operatorname{negl}(\lambda)\ \operatorname{tr}\left((3)+(4)\right).
\end{aligned}$$

The second line follows from the Pauli partitioning by Clifford (lemma 3.5). The third line holds by the linearity of trace and the fact that (3), (4) are orthogonal. This establishes the statistical indistinguishability between (1) + (2) and (3) + (4).

Step 4. Finally, we construct $\widetilde{\mathsf{Setup}}$, $\widetilde{\mathsf{Enc}}$, $\widetilde{\mathsf{Audit}}$ and prove receiver security.

```
$\widetilde{\mathsf{Setup}}(1^\lambda, 1^\ell)$:
1. Sample a random string $\mathsf{sk} \leftarrow \mathbb{Z}_p^{2\ell+3\lambda}$.
2. Output $(\mathsf{sk}, \phi_S, \sigma)$, where $(\sigma^{\hat{M}}, {\phi_S}^{\hat{S},\hat{C}})$ is the result of applying ${\mathsf{TP}^\dagger}^{\hat{M},\hat{S}}$ to $|\mathsf{sk}\rangle^{\hat{M},\hat{S},\hat{C}}$.
```

```
$\widetilde{\mathsf{Enc}}(\rho)$:
1. Sample a random Pauli $\mathsf{dk} \leftarrow \mathscr{P}_\ell$.
2. Output $(\mathsf{dk}, \sigma)$ where $\sigma = \mathsf{dk}(\rho)\mathsf{dk}^\dagger$.
```

```
$\widetilde{\mathsf{Audit}}(\mathsf{sk}, \mathsf{pf})$:
1. Parse $\mathsf{pf} = (\hat{P}, \hat{c})$.
2. Output $\perp$ if $(z(\hat{P}), x(\hat{P}), \hat{c}) \neq \mathsf{sk}$.
```

It is direct to see that $\mathsf{Recv} \circ \widetilde{\mathsf{Enc}} = \mathbb{1}$. We now analyze the state (dk, ϕ_A) that results from executing $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \mathcal{A})$ in a row and replacing $\mathsf{dk} \leftarrow \perp$ if $\perp \leftarrow \widetilde{\mathsf{Audit}}$. The execution of $\widetilde{\mathsf{Setup}}$ and $\widetilde{\mathsf{Enc}}$ yields random classical keys $\mathsf{sk} = r$, $\mathsf{dk} = P$ and quantum state

$$\left(\sigma^{\hat{M}}, {\phi_S}^{\hat{S},\hat{C}}, {\phi_R}^{\hat{R}}\right) = \mathsf{Mixed}\left[\left({\mathsf{TP}^\dagger}^{\hat{M},\hat{S}} |r\rangle^{\hat{M},\hat{S},\hat{C}}, P\rho^{\hat{R}}\right)\right]$$

After the execution of $\mathcal{A}$, the joint state is

$$\begin{aligned} &\mathop{\mathbb{E}}_{r,P} \mathsf{Mixed}\left[A\left({\mathsf{TP}^\dagger}^{\hat{M},\hat{S}} |r\rangle^{\hat{M},\hat{S},\hat{C}}, P\rho^{\hat{R}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right) \otimes |r\rangle \otimes |P\rangle\right] \\ = &\mathop{\mathbb{E}}_{r,P} \mathsf{Mixed}\left[A'\left(|r\rangle^{\hat{M},\hat{S},\hat{C}}, P\rho^{\hat{R}}, {\rho_{\mathsf{aux}}}^{\hat{W}}\right) \otimes |r\rangle \otimes |P\rangle\right] \end{aligned}$$

The next step is to apply $\widetilde{\mathsf{Audit}}$, which corresponds to the projection $|r\rangle\langle r|^{\hat{M},\hat{S},\hat{C}}$. The state conditioned that $\widetilde{\mathsf{Audit}}$ accepts can be analyzed through Pauli twirl with target measurement result 0 (lemma 3.3), which yields exactly (1). Similarly, the state conditioned that $\widetilde{\mathsf{Audit}}$ rejects is exactly (2). Hence, (dk, ϕ_A) generated from $(\mathsf{Setup}, \mathsf{Enc}, \mathcal{A}, \mathsf{Audit})$ and from $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Enc}}, \mathcal{A}, \widetilde{\mathsf{Audit}})$ are indistinguishable, which establishes receiver security.

□

6 MPQC-PVIA with Trusted Setup

Here, we present our MPQC protocol with a trusted setup. We make use of the Clifford-form AQA developed in the previous section together with cMPC to achieve PVIA security. For simplicity, we work with qubits i.e., $p = 2$ here. The protocol is divided into two parts:

1. An offline setup: a setup Σ^{PVIA} prepares EPR pairs $(\mathcal{S}_i, \mathcal{R}_i)$ of length ℓ_i and distributes the sending side $\mathcal{S}_i$ to party P_i. Next, the trusted setup encodes all of the receiving sides $\mathcal{R}_1, \cdots, \mathcal{R}_n$ and the ancilla ϕ_{anc} into a single ciphertext σ, which is sent to the server. This server can be any participant in the MPQC game; for simplicity, we can assume it's the first party. The above quantum states will later be utilized in the input encoding stage. The trusted setup then executes AQA.Setup to obtain portals for the output delivery stage. Finally, the trusted setup transmits information related to the secret keys to cMPC.

2. An online phase: every party acts as a client who teleports their input to the server. Directed by cMPC, the server evaluates the circuit on the ciphertext σ. Finally, the server returns the outputs to all clients using AQA.

Protocol 1 $(\Sigma^{\mathsf{PVIA}}, \Pi^{\mathsf{PVIA}})$ for MPQC-PVIA with Trusted Setup

Common Input: A quantum circuit C in the format of Section 4.

Trusted Setup Σ^{PVIA}**:**

1. Prepare EPR pairs $(\mathcal{S}_i{}^{S_i}, \mathcal{R}_i{}^{R_i})$ with $|S_i| = |R_i| = \ell_i$.
2. Initialize the ancilla register $N = (N_1, \cdots, N_d)$ as ϕ_{anc}.
3. Initialize the trap register $T = (T_1, \cdots, T_{n+d})$ as $|0\rangle^{\otimes(n+d)\lambda}$.
4. Sample $(\mathsf{sk}_i, \phi_{S,i}, \phi_{R,i}) \leftarrow \mathsf{AQA.Setup}(1^\lambda, 1^{\ell_i})$.
5. Sample random Clifford $E \leftarrow \mathscr{C}_{\ell_{\text{total}}+(n+d)\lambda}$ and apply $E^{R_1,\cdots,R_n,N,T}$ with result σ.
6. Send $(\sigma^{R_1,\cdots,R_n,N,T}, \phi_{S,1}, \cdots, \phi_{S,n})$ to server and send $(\mathcal{S}_i{}^{S_i}, \phi_{R,i})$ to client i.
7. Send the secrets $(E, \mathsf{sk}_1, \cdots, \mathsf{sk}_n)$ to cMPC.

Online Input: Client i receives $\rho_i \in \mathcal{D}^{\ell_i}$.

Protocol Π^{PVIA}**:**

Input Encoding:

1. Client i teleports ρ_i via $\mathcal{S}_i{}^{S_i}$ and sends the teleportation result P_i to cMPC.
2. cMPC sets the key $E_d = E^{R_1,\cdots,R_n,N,T} P_1{}^{R_1} \cdots P_n{}^{R_n}$.

Computation:

3. For $k = d, \cdots, 1$:
   (a) cMPC sends the gate $V_k = (E'_{k-1}{}^{R_{[n]},N_{[k-1]},T_{[n+k-1]}} \otimes P'_k \mathsf{CX}_{c_k}{}^{N_k,T_{n+k}}) E_k^{\dagger}{}^{R_{[n]},N_{[k]},T_{[n+k]}}$ to the server using a random Clifford $E'_{k-1} \leftarrow \mathscr{C}_{\Sigma_i \ell_i+(k-1)+(n+k-1)\lambda}$, a random Pauli $P'_k \leftarrow \mathscr{P}^*_{1+\lambda}$, and a random string $c_k \leftarrow \{0,1\}^\lambda$.
   (b) Server applies V_k to registers $(R_{[n]}, N_{[k]}, T_{[n+k]})$, measures (N_k, T_{n+k}) in the computational basis and sends the measurement outcome $r_k \in \{0,1\}^{1+\lambda}$ to cMPC.
   (c) cMPC sets $b_k \in \{0,1\}$ as the solution to $r_k \oplus x(P'_k) = b_k(1, c_k)$ if there is a solution. Otherwise, cMPC publicly outputs the server as malicious and aborts.
   (d) cMPC computes $G_{k-1}{}^{R_{[n]},N_{[k-1]}} = f_k(b_k, \cdots, b_d)$ and sets the key $E_{k-1} = E'_{k-1} G^{\dagger}_{k-1}$.

Output Delivery:

4. cMPC sends server $V' = \mathsf{AQA.EncG}(\mathsf{sk}_1)^{R_1,T_1} \cdots \mathsf{AQA.EncG}(\mathsf{sk}_n)^{R_n,T_n} E_0^{\dagger}{}^{R_1,\cdots,R_n,T_1,\cdots,T_n}$.
5. Server applies $V'^{R_1,\cdots,R_n,T_1,\cdots,T_n}$ and obtains $(\hat{\sigma}_1^{\hat{M}_1}, \cdots, \hat{\sigma}_n^{\hat{M}_n})$ where $\hat{M}_i := (R_i, T_i)$.

6. Server computes $\mathsf{pf}_i \leftarrow \mathsf{AQA.Send}(\hat{\sigma}_i, \phi_{S,i})$ and sends the result to cMPC.
7. cMPC computes $\mathsf{dk}_i \leftarrow \mathsf{AQA.Audit}(\mathsf{sk}_i, \mathsf{pf}_i)$. If $\mathsf{dk}_i = \bot$, cMPC publicly outputs the server as malicious and aborts. Otherwise, cMPC sends dk_i to client i.
8. Client i outputs $\rho'_i \leftarrow \mathsf{AQA.Recv}(\mathsf{dk}_i, \phi_{R,i})$.

6.1 Security

Theorem 6.1. $(\Sigma^{\mathsf{PVIA}}, \Pi^{\mathsf{PVIA}})$ *is a multi-party quantum computation secure with publicly verifiable identifiable abort with trusted setup in the MPC-hybrid model as defined in Definition 4.2. i.e., For every non-uniform (*QPT*) adversary $\mathcal{A}$ corrupting party set I with $|I| \le n-1$, there is a non-uniform (*QPT*) adversary $\mathsf{Sim}_{\mathcal{A}}$ corrupting I, such that for any (possibly entangled) states $\rho_1, \cdots \rho_n, \rho_{\mathsf{aux}}$,*

$$\{\mathsf{Real}^{\Pi^{\mathsf{PVIA}} \circ \Sigma^{\mathsf{PVIA}}}_{\mathcal{A}(\rho_{\mathsf{aux}})}(1^\lambda, C, \rho_1, \cdots, \rho_n)\} \approx \{\mathsf{Ideal}^{\mathsf{MPQC}}_{\mathsf{Sim}_{\mathcal{A}}(\rho_{\mathsf{aux}})}(1^\lambda, 0, C, \rho_1, \cdots, \rho_n)\}$$

Proof. Consider the following hybrid worlds modified from the real protocol gradually. We describe each hybrid in terms of the changes made to the previous hybrid.

- $\mathcal{H}_1$: Introduce a trusted party $\mathsf{T}^{\mathsf{hybrid}}$ who executes both the setup Σ^{PVIA} and cMPC.
- $\mathcal{H}_2$:
  - Σ^{PVIA} step 1: $\mathsf{T}^{\mathsf{hybrid}}$ prepares EPR pairs on registers $(S_i, \tilde{R}_i), (\tilde{S}_i, R_i)$ with $|S_i| = |\tilde{R}_i| = |\tilde{S}_i| = |R_i| = \ell_i$ and keeps $\{(\tilde{R}_i, \tilde{S}_i)\}$. Let $\tilde{\mathcal{R}}_i$ be the content of $\tilde{R}_i$.
  - Σ^{PVIA} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ prepares EPR pairs on (D, N) of length d and keeps D.
  - Π^{PVIA} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ extracts the input $\tilde{\rho}_i^{\tilde{R}_i} = P_i^\dagger \tilde{\mathcal{R}}_i^{\tilde{R}_i}$ and prepares ancilla ${\phi_{\mathsf{anc}}}^{\tilde{N}}$. Next, it teleports $\tilde{\rho}_i^{\tilde{R}_i}, {\phi_{\mathsf{anc}}}^{\tilde{N}}$ back to the server via registers $\tilde{S}_i, D$ respectively. If the teleportation results are $\tilde{P}_i, \tilde{P}_{\mathsf{anc}}$, it sets $E_d = E^{R_1, \cdots, R_n, N, T} \tilde{P}_1^{R_1} \cdots \tilde{P}_n^{R_n} \tilde{P}_{\mathsf{anc}}^{N}$.
- $\mathcal{H}_{3+d-k}$, where $k = d, \cdots, 1$:
  - Σ^{PVIA} step 5: For $h = d, \cdots, k$, $\mathsf{T}^{\mathsf{hybrid}}$ resets (N_h, T_{n+h}) as random string $\tilde{r}_h \leftarrow \{0,1\}^{\lambda+1}$, samples random Clifford $V_h \leftarrow \mathscr{C}_{\ell^{\text{total}}+(n+h)\lambda}$, $\tilde{E}_{k-1} \leftarrow \mathscr{C}_{\ell^{\text{total}}+(n+k-1)\lambda}$ and sets $E = V_d^{\dagger\, R_{[n]}, N_{[d]}, T_{[n+d]}} \cdots V_k^{\dagger\, R_{[n]}, N_{[k]}, T_{[n+k]}} \tilde{E}_{k-1}^{R_{[n]}, N_{[k-1]}, T_{[n+k-1]}}$.
  - Π^{PVIA} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ computes $C[k] \cdots C[d] \left(\tilde{\rho}_1^{\tilde{R}_1}, \cdots, \tilde{\rho}_n^{\tilde{R}_n}, {\phi_{\mathsf{anc}}}^{\tilde{N}}\right)$. Next, it teleports the states in registers $\tilde{R}_i$ and $\tilde{N}'_{[k-1]}$ back to the server. If the teleportation results are $\tilde{P}_i, \tilde{P}_{\mathsf{anc}}$, it sets $E_{k-1} = \tilde{E}_{k-1}^{R_1, \cdots, R_n,, N_{[k-1]}, T_{[n+k-1]}} \tilde{P}_1^{R_1} \cdots \tilde{P}_n^{R_n} \tilde{P}_{\mathsf{anc}}^{N_{[k-1]}}$.
  - Π^{PVIA} step 3 iteration h for $h = d, \cdots, k$: $\mathsf{T}^{\mathsf{hybrid}}$ sends V_h to the server and receives r_h in return. If $r_h \neq \tilde{r}_h$, $\mathsf{T}^{\mathsf{hybrid}}$ publicly outputs the server as malicious and aborts.
- $\mathcal{H}_{3+d}$:

- Σ^{PVIA} step 4: $\mathsf{T}^{\mathsf{hybrid}}$ samples $(\mathsf{sk}_i, \phi_{S,i}, \sigma_i) \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Setup}}(1^\lambda, 1^{\ell_i})$, uses the content $\mathcal{R}_i$ of register R_i as $\phi_{R,i}$, and reassigns register (R_i, T_i) as σ_i.
- Π^{PVIA} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ computes the circuit C with input $(\tilde{\rho}_1^{\tilde{R}_1}, \cdots \tilde{\rho}_n^{\tilde{R}_n})$, teleports the output via register $\tilde{S}_i$, and sets dk_i as the teleportation result $\tilde{P}_i$.
- Π^{PVIA} step 5: $\mathsf{T}^{\mathsf{hybrid}}$ sets $V' = \tilde{E}_0^\dagger$.
- Π^{PVIA} step 7: $\mathsf{T}^{\mathsf{hybrid}}$ resets $\mathsf{dk}_i = \perp$ if $\perp \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Audit}}(\mathsf{sk}_i, \mathsf{pf}_i)$.

The last hybrid world is equivalent to the ideal world with the following simulator.

Simulator 1 $\mathsf{Sim}^{\mathsf{PVIA}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for MPQC-PVIA with Trusted Setup

1. Fake setup (if client i is corrupted):
   (a) Prepare EPR pairs on registers $(S_i, \tilde{R}_i)$ with $|S_i| = |\tilde{R}_i| = \ell_i$.
   (b) Prepare EPR pairs on registers $(\tilde{S}_i, \hat{R}_i)$ with $|\tilde{S}_i| = |\hat{R}_i| = \ell_i$.
   (c) Send $(S_i, \hat{R}_i)$ to $\mathcal{A}$.
2. Fake setup (if server is corrupted):
   (a) Sample $(\mathsf{sk}_i, \phi_{S,i}, \sigma_i{}^{R_i,T_i}) \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Setup}}(1^\lambda, 1^{\ell_i})$ for $i \in [n]$.
   (b) Initialize registers (N_k, T_{n+k}) with random $\tilde{r}_k \leftarrow \{0,1\}^{\lambda+1}$ for $k \in [d]$.
   (c) Sample random Clifford gates $V_k \leftarrow \mathscr{C}_{\ell^{\mathrm{total}}+(n+k)\lambda}$ for $k = 0, 1, \cdots, d$.
   (d) Apply $V_d^{\dagger R_{[n]},N_{[d]},T_{[n+d]}} \cdots V_1^{\dagger R_{[n]},N_{[1]},T_{[n+1]}} V_0^{\dagger R_{[n]},T_{[n]}}$ with resulting state σ.
   (e) Send $(\sigma^{R_1,\cdots,R_n,N,T}, \phi_{S,1}, \cdots, \phi_{S,n})$ to $\mathcal{A}$.
3. Input extraction: Receive P_i from $\mathcal{A}$ and extract the input $\tilde{\rho}_i^{\tilde{R}_i} = P_i^\dagger \tilde{\mathcal{R}}_i^{\tilde{R}_i}$.
4. Invoke the ideal functionality: Send $\tilde{\rho}_i$ to T and receive the output $\tilde{\rho}'_i$ from T.
5. Check the abort decision (if server is corrupted):
   (a) For $k = d, \cdots, 1$, send V_k to $\mathcal{A}$ and receive r_k in return.
   If $r_k \neq \tilde{r}_k$, send `abort` to T in the name of the server.
   (b) Send V_0 to $\mathcal{A}$.
   (c) For $i \in [n]$, receive pf_i from $\mathcal{A}$.
   If $\perp \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Audit}}(\mathsf{sk}_i, \mathsf{pf}_i)$, send `abort` to T in the name of the server.
6. Output delivery: Teleport $\tilde{\rho}'_i$ via $\tilde{S}_i$, obtain its result $\tilde{P}_i$ and send $\mathsf{dk}_i = \tilde{P}_i$ to $\mathcal{A}$ if the server did not send `abort` in the previous step.
7. Output $\mathcal{A}$'s output.

To prove Theorem 6.1, it suffices to show indistinguishability between consecutive hybrids. When comparing consecutive hybrids, we will view the execution of the adversary and the protocol until a certain step as a quantum operation, which is a sum of CP maps of the form $\tau \mapsto A\tau A^\dagger$. Then, it suffices to show indistinguishability under every such CP map because indistinguishability of subnormalized states extend under addition.

Note that the indistinguishability holds whenever the adversary aborts the classical MPC. The reason is that the abort prevents cMPC from producing information related to the Pauli or Clifford keys, so the quantum states are all maximally mixed according to Pauli twirl and Clifford twirl (lemma 3.2). Hence, we will assume that the adversary does not abort cMPC from now on. Below, we sometimes abbreviate parts of the state that are currently irrelevant with dots and omit writing the expectation over currently irrelevant keys.

- $\mathsf{Real} = \mathcal{H}_1$: This is because Σ^{PVIA}, cMPC and $\mathsf{T}^{\mathsf{hybrid}}$ are all trusted executions.

- $\mathcal{H}_1 = \mathcal{H}_2$: Consider executing the protocol until the end of step 1. Suppose the CP map A projects $(M_{[n]}, S_{[n]})$ to $|P_{[n]}\rangle$ in step 1. The state of $\mathcal{H}_1$ after protocol step 2 is

$$\begin{aligned} &\mathop{\mathbb{E}}_{E \leftarrow \mathscr{C}} \mathsf{Mixed}\left[A\left(E\left({\mathcal{R}_{[n]}}^{R_{[n]}}, {\phi_{\mathsf{anc}}}^{N}, |0\rangle^{\otimes(n+d)\lambda^{T}}\right), \cdots\right) \otimes |E_d\rangle\right] \\ = &\mathop{\mathbb{E}}_{E_d \leftarrow \mathscr{C}} \mathsf{Mixed}\left[A\left(E_d\left(P_{[n]}^{\dagger}\mathcal{R}_{[n]}{}^{R_{[n]}}, {\phi_{\mathsf{anc}}}^{N}, |0\rangle^{\otimes(n+d)\lambda^{T}}\right), \cdots\right) \otimes |E_d\rangle\right] \end{aligned}$$

  In $\mathcal{H}_2$, we can change the order of A and the teleportation made by $\mathsf{T}^{\mathsf{hybrid}}$ because these operators act on disjoint registers. By quantum teleportation, the state of $\mathcal{H}_2$ after protocol step 2 is as follows, which is the same as in $\mathcal{H}_1$.

$$\begin{aligned} &\mathop{\mathbb{E}}_{E \leftarrow \mathscr{C}} \mathsf{Mixed}\left[A\left(E\left(\tilde{P}_{[n]}P_{[n]}^{\dagger}\tilde{\mathcal{R}}_{[n]}{}^{R_{[n]}}, \tilde{P}_{\mathsf{anc}}\phi_{\mathsf{anc}}{}^{N}, |0\rangle^{\otimes(n+d)\lambda^{T}}\right), \cdots\right) \otimes |E_d\rangle\right] \\ = &\mathop{\mathbb{E}}_{E_d \leftarrow \mathscr{C}} \mathsf{Mixed}\left[A\left(E_d\left(P_{[n]}^{\dagger}\tilde{\mathcal{R}}_{[n]}{}^{R_{[n]}}, {\phi_{\mathsf{anc}}}^{N}, |0\rangle^{\otimes(n+d)\lambda^{T}}\right), \cdots\right) \otimes |E_d\rangle\right] \end{aligned}$$

- $\mathcal{H}_{2+d-k} \approx \mathcal{H}_{3+d-k}$ for $k = d, \cdots, 1$: Consider executing the protocol until the end of step 3 iteration $k{+}1$. Suppose the CP map A projects $(M_{[n]}, S_{[n]})$ to $|P_{[n]}\rangle$ in step 1 and projects $N_d, \cdots, N_{k+1}$ to $b_d, \cdots, b_{k+1}$ in step 3 for iterations d to $k+1$. Let us denote the partial computation result $C[k{+}1]\cdots C[d]\big(P_{[n]}^{\dagger}\tilde{\mathcal{R}}_{[n]}, \phi_{\mathsf{anc}}\big)$ as $\tilde{\rho}_{(k)}{}^{\tilde{R}_{[n]},\tilde{N}_{[k]}} = \sum_b \tilde{\rho}_{(k,b)}{}^{\tilde{R}_{[n]},\tilde{N}_{[k-1]}} \otimes |b\rangle^{\tilde{N}_k}$. In $\mathcal{H}_{2+d-k}$, the state at protocol step $3(a)$ for iteration k is

$$\begin{aligned} &\mathop{\mathbb{E}}_{E_k, E'_{k\text{-}1}, P'_k, c_k} \mathsf{Mixed}\left[A\left(E_k\left(\tilde{\rho}_{(k)}{}^{R_{[n]},N_{[k]}}, |0\rangle^{\otimes(n+k)\lambda^{T_{[n+k]}}}\right), \cdots\right) \otimes |P'_k, c_k\rangle\right] \\ = &\mathop{\mathbb{E}}_{V_k, E'_{k\text{-}1}, P'_k, c_k} \mathsf{Mixed}\left[A\left(V_k^{\dagger}\left(\sum_b E'_{k-1}\big(\tilde{\rho}_{(k,b)}, |0\rangle^{\otimes(n+k-1)\lambda}\big) \otimes P'_k\mathsf{CX}_{c_k}\big(|b\rangle, |0\rangle^{\otimes\lambda}\big)\right), \cdots\right) \otimes |P'_k, c_k\rangle\right] \end{aligned}$$

  The equality follows from the definition of V_k. Since the protocol gives the attacker access to $|V_k\rangle$ in step $3(a)$, we can merge $V_k^{\dagger}$ and A into a CP map A' that operates also on $|V_k\rangle$.

Also, we can simplify $\mathsf{CX}_{c_k}(b, |0\rangle^{\otimes\lambda}) = |b(1, c_k)\rangle$. Thus, the state is eqaul to

$$\mathop{\mathbb{E}}_{E'_{k\text{-}1}, P'_k, c_k} \mathsf{Mixed}\left[A'\left(\sum_b E'_{k-1}\big(\tilde{\rho}_{(k,b)}, |0\rangle^{\otimes(n+k-1)\lambda}\big) \otimes P'_k\, |b(1, c_k)\rangle^{N_k, T_{n+k}}, \cdots\right) \otimes |P'_k, c_k\rangle\right]$$

Afterwards, protocol step 3(c) projects registers (N_k, T_{n+k}) to $|x(P'_k) + b_k(1, c_k)\rangle$ for $b_k \in \{0, 1\}$. The two cases are similar, and we demonstrate using the case $b_k = 0$. To analyze, we apply Pauli twirl with target measurement result $(0, 0^\lambda)$ (lemma 3.3) and use the Pauli decomposition $A' = \sum_{Q \in \mathscr{P}^*_{\lambda+1}} Q^{N_k, T_{n+k}} \otimes A'_Q$. The state after the projection of step 3(c) for obtaining solution $b_k = 0$ is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{E'_{k\text{-}1}, P'_k, c_k} \mathsf{Mixed}\left[|x(P'_k)\rangle\langle x(P'_k)|^{N_k, T_{n+k}} A'\left(\sum_b P'_k\, |b(1, c_k)\rangle^{N_k, T_{n+k}} \otimes E'_{k-1}\big(\tilde{\rho}_{(k,b)}, |0\rangle^{\otimes(n+k-1)\lambda}\big), \cdots\right)\right] \\
&= \mathop{\mathbb{E}}_{E'_{k\text{-}1}, \tilde{r}_k, c_k} \left(\mathsf{Mixed}\left[\sum_{x(Q) = (0, 0^\lambda)} Q\, |\tilde{r}_k\rangle \otimes A'_Q \left(E'_{k-1}\big(\tilde{\rho}_{(k,0)}, |0\rangle^{\otimes(n+k-1)\lambda}\big), \cdots\right)\right]\right. \\
&\qquad\qquad \left. + \ \mathsf{Mixed}\left[\sum_{x(Q) = (1, c_k)} Q\, |\tilde{r}_k\rangle \otimes A'_Q \left(E'_{k-1}\big(\tilde{\rho}_{(k,1)}, |0\rangle^{\otimes(n+k-1)\lambda}\big), \cdots\right)\right]\right) \\
&\approx \mathop{\mathbb{E}}_{E'_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[\sum_{x(Q) = (0, 0^\lambda)} Q\, |\tilde{r}_k\rangle \otimes A'_Q \left(E'_{k-1}\big(\tilde{\rho}_{(k,0)}, |0\rangle^{\otimes(n+k-1)\lambda}\big), \cdots\right)\right]
\end{aligned}$$

The indistinguishability follows because each c_k occurs with negligible probability. After setting the key $E_{k-1} = E'_{k-1} G^\dagger_{k-1}$ in protocol step 3(k), the state in $\mathcal{H}_{2+d-k}$ becomes

$$\begin{aligned}
&\mathop{\mathbb{E}}_{E_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[\sum_{x(Q) = (0, 0^\lambda)} Q\left(|\tilde{r}_k\rangle\right) \otimes A'_Q \left(E_{k-1}\left(G_{k-1}\big(\tilde{\rho}_{(k,0)}\big), |0\rangle^{\otimes(n+k-1)\lambda}\right), \cdots\right)\right] \\
&= \mathop{\mathbb{E}}_{E_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[\sum_{x(Q) = (0, 0^\lambda)} Q\left(|\tilde{r}_k\rangle\right) \otimes A'_Q \left(E_{k-1}\left(\tilde{\rho}_{(k-1)}, |0\rangle^{\otimes(n+k-1)\lambda}\right), \cdots\right)\right]
\end{aligned}$$

In $\mathcal{H}_{3+d-k}$, the state of acceptance after projecting to $b_k = 0$ is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{E_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[|\tilde{r}_k\rangle\langle\tilde{r}_k|^{N_k, T_{n+k}} A' \left(|\tilde{r}_k\rangle^{N_k, T_{n+k}}, E_{k-1}\left(\tilde{\rho}_{(k-1)}, |0\rangle^{\otimes(n+k-1)\lambda}\right), \cdots\right)\right] \\
&= \mathop{\mathbb{E}}_{E_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[\sum_Q |\tilde{r}_k\rangle\langle\tilde{r}_k|\, Q\, |\tilde{r}_k\rangle \otimes A'_Q \left(E_{k-1}\left(\tilde{\rho}_{(k-1)}, |0\rangle^{\otimes(n+k-1)\lambda}\right), \cdots\right)\right] \\
&= \mathop{\mathbb{E}}_{E_{k-1}, \tilde{r}_k} \mathsf{Mixed}\left[\sum_{x(Q) = (0, 0^\lambda)} Q\left(|\tilde{r}_k\rangle\right) \otimes A'_Q \left(E_{k-1}\left(\tilde{\rho}_{(k-1)}, |0\rangle^{\otimes(n+k-1)\lambda}\right), \cdots\right)\right]
\end{aligned}$$

Thus, the states in $\mathcal{H}_{2+d-k}$ and $\mathcal{H}_{3+d-k}$ are indistinguishable following the projections for both $b_k \in \{0, 1\}$. When both projections fail, $\mathsf{T}^{\mathsf{hybrid}}$ aborts in both $\mathcal{H}_{2+d-k}$ and $\mathcal{H}_{3+d-k}$ by publicly outputting the server P_1 as malicious and will not use the Clifford keys anymore. By Clifford twirl (lemma 3.2), the states in both $\mathcal{H}_{2+d-k}$ and $\mathcal{H}_{3+d-k}$ will be indistinguishable to

$$\mathop{\mathbb{E}}_{\tilde{r}_k, \tilde{r}_k^*} \mathsf{Mixed}\left[\sum_{x(Q) \neq (0, 0^\lambda)} Q\left(|\tilde{r}_k\rangle\right)^{N_k, T_{n+k}} \otimes A'_Q\left(|\tilde{r}^*\rangle^{N_{[k-1]}, T_{[n+k-1]}}, \cdots\right) \otimes |\bot_{\mathsf{P}_1}\rangle\right]$$

- $\mathcal{H}_{2+d} \approx \mathcal{H}_{3+d}$: Consider executing the protocol until the end of step 6. Let $(\tilde{\rho}'_1, \cdots, \tilde{\rho}'_n)$ be the quantum output of C. The state of $\mathcal{H}_{2+d}$ at step 6 is

$$\begin{aligned} & \mathop{\mathbb{E}}_{E_0, \mathsf{sk}_i} \mathsf{Mixed}\left[A\left({E_0}^{R_1, \cdots, R_n, T_1, \cdots, T_n}\left({\tilde{\rho}'_i}^{R_i}, |0\rangle^{\otimes \lambda T_i}\right), |V'\rangle, \phi_{S,i}, \phi_{R,i}, \rho_{\mathsf{aux}}\right) \otimes |\mathsf{sk}_i\rangle\right] \\ = & \mathop{\mathbb{E}}_{V', \mathsf{sk}_i} \mathsf{Mixed}\left[A\left({V'^\dagger}^{R_1, \cdots, R_n, T_1, \cdots, T_n}\left(\mathsf{AQA.EncG}(\mathsf{sk}_i)\left(\tilde{\rho}'_i, |0\rangle^{\otimes \lambda}\right)^{R_i, T_i}\right), |V'\rangle, \phi_{S,i}, \phi_{R,i}, \rho_{\mathsf{aux}}\right) \otimes |\mathsf{sk}_i\rangle\right] \\ = & \mathop{\mathbb{E}}_{V', \mathsf{sk}_i} \mathsf{Mixed}\left[A\left({V'^\dagger}^{R_1, \cdots, R_n, T_1, \cdots, T_n}\left(\mathsf{AQA.Enc}\left(\mathsf{sk}_i, \tilde{\rho}'_i\right)^{R_i, T_i}\right), |V'\rangle, \phi_{S,i}, \phi_{R,i}, \rho_{\mathsf{aux}}\right) \otimes |\mathsf{sk}_i\rangle\right] \end{aligned}$$

  where the first equality follows from the definition of V' in $\mathcal{H}_{2+d}$ and the second follows from the definition of AQA.Enc. After protocol step 7, the state of $\mathcal{H}_{2+d}$ can be generated under the process (AQA.Setup, AQA.Enc, A', Audit) where A' is a CP map that merges A and $V'^\dagger$. On the other hand, the state of $\mathcal{H}_{3+d}$ at step 6 is

$$\begin{aligned} & \mathop{\mathbb{E}}_{\tilde{E}_0, \mathsf{sk}_i, \tilde{P}_i} \mathsf{Mixed}\left[A\left(\tilde{E}_0^{R_1, \cdots, R_n, T_1, \cdots, T_n}\left({\sigma_i}^{R_i, T_i}\right), |V'\rangle, \phi_{S,i}, \tilde{P}_i(\tilde{\rho}'_i), \rho_{\mathsf{aux}}\right) \otimes |\tilde{P}_i\rangle \otimes |\mathsf{sk}_i\rangle\right] \\ = & \mathop{\mathbb{E}}_{V', \mathsf{sk}_i, \mathsf{dk}_i} \mathsf{Mixed}\left[A\left({V'^\dagger}^{R_1, \cdots, R_n, T_1, \cdots, T_n}\left({\sigma_i}^{R_i, T_i}\right), |V'\rangle, \phi_{S,i}, \mathsf{dk}_i(\tilde{\rho}'_i), \rho_{\mathsf{aux}}\right) \otimes |\mathsf{dk}_i\rangle \otimes |\mathsf{sk}_i\rangle\right] \end{aligned}$$

  where the equality follows from the definition of V' and dk_i in $\mathcal{H}_{3+d}$. Here, $\mathsf{sk}_i, \phi_{S,i}, \sigma_i$ are generated from AQA.$\widetilde{\mathsf{Setup}}$, whereas dk_i and the encoding of $\tilde{\rho}'_i$ can be generated from AQA.$\widetilde{\mathsf{Enc}}$. After protocol step 7, the state of $\mathcal{H}_{3+d}$ can be generated under the process (AQA.$\widetilde{\mathsf{Setup}}$, AQA.$\widetilde{\mathsf{Enc}}$, A', $\widetilde{\mathsf{Audit}}$). By the receiver security of AQA (Theorem 5.2), these two states are statistically indistinguishable.

- $\mathcal{H}_{3+d} =$ Ideal: Except for the computation of C, the execution of $\mathsf{T}^{\mathsf{hybrid}}$ in $\mathcal{H}_{3+d}$ consists of n independent parts, each interacting with only one party. Thus, we can collect the parts that interact with the corrupted parties as a simulator Sim, who also interacts with the ideal functionality of MPQC to compute C. By quantum teleportation and Theorem 5.2, the honest parties and their corresponding parts of $\mathsf{T}^{\mathsf{hybrid}}$ indeed send the honest inputs to the ideal functionality, never send `abort` to the ideal functionality, and always output the honest outputs. Hence, $\mathcal{H}_{3+d}$ is identical to Ideal.

□

7 BoBW-MPQC-PVIA with Trusted Setup

In this section, we construct a best-of-both-worlds multi-party quantum computation protocol secure with publicly verifiable identifiable abort (BoBW-MPQC-PVIA). The protocol is similar to the MPQC-PVIA protocol but with n servers instead. P_j will be assigned as server j. The protocol is divided into two parts:

1. An offline setup: a setup Σ^{BoBW} prepares EPR pairs $(\mathcal{S}_i, \mathcal{R}_i)$ of length ℓ_i and distributes the sending side $\mathcal{S}_i$ to party P_i. Next, the setup uses a QECC scheme to encode every $\mathcal{R}_i$ into $\{\mathcal{R}_i^{(j)}\}_{j\in[n]}$ and the ancilla ϕ_{anc} into $\{\phi_{\mathsf{anc}}^{(j)}\}_{j\in[n]}$. Afterwards, the setup encrypts the j-th part of the QECC codewords into a Clifford ciphertext and send it to server j. The setup then executes $\mathsf{AQA.Setup}$ to prepare for the output delivery stage. Finally, the setup sends the secret keys to cMPC.

2. An online phase: every party acts as a client who teleports their input to the servers. Directed by cMPC, the servers evaluate the circuit in a fault-tolerant manner. Each server j operates only on the ciphertext that encrypts the j-th part of the QECC codewords. Finally, the servers returns the outputs to the clients using AQA.

Let $t < \frac{n}{2}$ and let QECC be a $[[n, 1]]_p$ polynomial code that corrects t erasure errors as in Lemma 3.7. Recall that QECC stands for the Clifford gate for QECC encoding. We present the protocol formally as follows:

Protocol 2 $(\Sigma^{\mathsf{BoBW}}, \Pi^{\mathsf{BoBW}})$ for BoBW-MPQC-PVIA with Trusted Setup

Common Input: A threshold t and a quantum circuit C in the format of Section 4.

Trusted Setup Σ^{BoBW}**:**

1. Prepare EPR pairs $({\mathcal{S}_i}^{S_i}, \mathcal{R}_i)$ of length ℓ_i. Compute $({\mathcal{R}_i^{(1)}}^{R_{1,i}}, \cdots, {\mathcal{R}_i^{(n)}}^{R_{n,i}}) \leftarrow \mathsf{QECC.Enc}(\mathcal{R}_i)$.
2. Initialize the ancilla $({\phi_{\mathsf{anc}}^{(1)}}^{N_1}, \cdots, {\phi_{\mathsf{anc}}^{(n)}}^{N_n}) \leftarrow \mathsf{QECC.Enc}(\phi_{\mathsf{anc}})$ where $N_j = (N_{j,1}, \cdots, N_{j,d})$.
3. Initialize the trap registers $T_j = (T_{j,1}, \cdots, T_{j,n+d})$ as $|0\rangle^{\otimes(n+d)\lambda}$.
4. Sample $(\mathsf{sk}_{j\to i},\ \phi_{S,j\to i},\ \phi_{R,j\to i}) \leftarrow \mathsf{AQA.Setup}(1^\lambda, 1^{\ell_i})$.
5. Sample random Clifford $E_j \leftarrow \mathscr{C}_{\ell_{\mathrm{total}}+(n+d)\lambda}$ and apply ${E_j}^{R_{j,1},\cdots,R_{j,n},N_j,T_j}$ with result σ_j.
6. Send $({\mathcal{S}_i}^{S_i}, {\sigma_i}^{R_{i,1},\cdots,R_{i,n},N_i,T_i}, \phi_{S,i\to 1}, \cdots, \phi_{S,i\to n}, {\phi_{R,1\to i}}^{\hat{R}_{1,i}}, \cdots, {\phi_{R,n\to i}}^{\hat{R}_{n,i}})$ to party P_i.
7. Send the secrets $\{E_j, \mathsf{sk}_{j\to i}\}_{i,j\in[n]}$ and the corruption list $J = \emptyset$ to cMPC.

Online Input: Party P_i receives ${\rho_i}^{M_i} \in \mathcal{D}^{\ell_i}$.

Protocol Π^{BoBW}**:**

Input Encoding:

1. Client i teleports ρ_i via ${\mathcal{S}_i}^{S_i}$ and sends the teleportation result P_i to cMPC.
2. cMPC computes the effective teleportation Pauli $P_{j,i} \in \mathscr{P}_{\ell_i}$ for each server j where $(P_{1,i} \otimes \cdots \otimes P_{n,i}) = \mathsf{QECC}\ P_i\ \mathsf{QECC}^\dagger$. Accordingly to the Pauli gates $(P_{j,1}, \cdots, P_{j,n})$, cMPC sets the key $E_{j,d} = {E_j}^{R_{j,1},\cdots,R_{j,n},N_j,T_j} {P_{j,i}}^{R_{j,1}} \cdots {P_{j,n}}^{R_{j,n}}$.

Computation:

3. For $k = d, \cdots, 1$ and for each server j:

   (a) cMPC sends $V_{j,k} = ({E'_{j,k-1}}^{R_{j,[n]},N_{j,[k-1]},T_{j,[n+k-1]}} \otimes P'_{j,k}\mathsf{CX}_{c_{j,k}}{}^{N_{j,k},T_{j,n+k}}){E^\dagger_{j,k}}^{R_{j,[n]},N_{j,[k]},T_{j,[n+k]}}$ to server j using a random Clifford $E'_{j,k-1} \leftarrow \mathscr{C}_{\Sigma_i \ell_i+(k-1)+(n+k-1)\lambda}$, a random Pauli $P'_{j,k} \leftarrow \mathscr{P}^*_{1+\lambda}$, and random $c_{j,k} \leftarrow \mathbb{Z}_p^\lambda$.

   (b) Server j applies $V_{j,k}$ to registers $(R_{j,[n]}, N_{j,[k]}, T_{j,[n+k]})$, measures $(N_{j,k}, T_{j,n+k})$ in the computational basis and sends the measurement outcome $r_{j,k} \in \mathbb{Z}_p^{1+\lambda}$ to cMPC.

   (c) cMPC sets $b_{j,k} \in \mathbb{Z}_p$ as the solution to $r_{j,k} \oplus x(P'_{j,k}) = b_{j,k}(1, c_{j,k})$ if there is a solution. Otherwise, cMPC adds server j into the corruption list J and will not interact with this server anymore.

   (d) If $|J| > t$, cMPC publicly outputs J as malicious and aborts.

   (e) cMPC computes the decoded measurement result $b_k \leftarrow \mathsf{QECC.Dec}((b_{1,k}, \cdots, b_{n,k}), J)$, the next gate $G_{k-1} = f_k(b_1, \cdots, b_k)$, and its fault-tolerant version $(G^{(1)}_{k-1}, \cdots, G^{(n)}_{k-1})$. Accordingly, cMPC sets the key $E_{j,k-1} = E'_{j,k-1}G^{(j)\dagger}_{k-1}$.

Output Delivery:

4. cMPC sends the gate $V'_j = \mathsf{AQA.EncG}(\mathsf{sk}_{j\to 1})^{R_{j,1},T_{j,1}} \cdots \mathsf{AQA.EncG}(\mathsf{sk}_{j\to n})^{R_{j,n},T_{j,n}} {E^\dagger_{j,0}}^{R_{j,[n]},T_{j,[n]}}$ to server j.

5. Server j applies ${V'_j}^{R_{j,1},\cdots,R_{j,n},T_{j,1},\cdots,T_{j,n}}$ and obtains $(\hat{\sigma}_{j,1}^{\hat{M}_{j,1}}, \cdots, \hat{\sigma}_{j,n}^{\hat{M}_{j,n}})$ where $\hat{M}_{j,i} := (R_{j,i}, T_{j,i})$.

6. Server j computes $\mathsf{pf}_{j,i} \leftarrow \mathsf{AQA.Send}(\hat{\sigma}_{j,i}, \phi_{S,j\to i})$ and sends the result to cMPC.

7. cMPC computes $\mathsf{dk}_{j,i} \leftarrow \mathsf{AQA.Audit}(\mathsf{sk}_{j\to i}, \mathsf{pf}_{j,i})$. If $\mathsf{dk}_{j,i} = \perp$, cMPC adds server j into the corruption list J. If $|J| > t$, cMPC publicly outputs J as malicious and aborts.

8. cMPC sends $\mathsf{dk}_{j,i}$ to client i.

9. Client i computes $\rho'_{j,i} \leftarrow \mathsf{AQA.Recv}(\mathsf{dk}_{j,i}, \phi_{R,j\to i})$ and outputs $\rho'_i \leftarrow \mathsf{QECC.Dec}((\rho'_{1,i}, \cdots \rho'_{n,i}), J)$.

7.1 Security

Theorem 7.1. *When parameterized by $t < \frac{n}{2}$, $(\Sigma^{\mathsf{BoBW}}, \Pi^{\mathsf{BoBW}})$ is a best-of-both-worlds multiparty quantum computation secure with publicly verifiable identifiable abort of threshold t with trusted setup in the MPC-hybrid model as defined in Definition 4.1. i.e., For every non-uniform (QPT) adversary $\mathcal{A}$ corrupting party set I with $|I| \leq n - t - 1$, there is a non-uniform (QPT) adversary $\mathsf{Sim}_\mathcal{A}$ corrupting I, such that for any (possibly entangled) states $\rho_1, \cdots \rho_n, \rho_{\mathsf{aux}}$,*

$$\{\mathsf{Real}^{\Pi^{\mathsf{BoBW}} \circ \Sigma^{\mathsf{BoBW}}}_{\mathcal{A}(\rho_{\mathsf{aux}})}(1^\lambda, t, C, \rho_1, \cdots, \rho_n)\} \approx \{\mathsf{Ideal}^{\mathsf{MPQC}}_{\mathsf{Sim}_\mathcal{A}(\rho_{\mathsf{aux}})}(1^\lambda, t, C, \rho_1, \cdots, \rho_n)\}$$

The proof of Theorem 7.1 bears resemblance to that of Theorem 6.1, with some additional analysis owing to the use of quantum error correction codes.

Proof. Consider the following hybrid worlds modified from the real protocol gradually. We describe each hybrid in terms of the changes made to the previous hybrid.

- $\mathcal{H}_1$: Introduce a trusted party $\mathsf{T}^{\mathsf{hybrid}}$ who executes both the setup Σ^{BoBW} and cMPC.
- $\mathcal{H}_2$:
  - Σ^{BoBW} step 1: $\mathsf{T}^{\mathsf{hybrid}}$ prepares EPR pairs on $(S_i, \tilde{R}_i), (\tilde{S}_{j,i}, R_{j,i})$ with $|S_i| = |\tilde{R}_i| = |\tilde{S}_{j,i}| = |R_{j,i}| = \ell_i$ and keeps $\{(\tilde{R}_i, \tilde{S}_{j,i})\}$. Let $\tilde{\mathcal{R}}_i$ be the content of $\tilde{R}_i$.
  - Σ^{BoBW} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ prepares EPR pairs on (D_j, N_j) of length d and keeps D_j.
  - Π^{BoBW} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ extracts the input $\tilde{\rho}_i^{\tilde{R}_i} = P_i^\dagger \tilde{\mathcal{R}}_i^{\tilde{R}_i}$, performs $(\tilde{\rho}_i^{(1)}, \cdots \tilde{\rho}_i^{(n)}) \leftarrow \mathsf{QECC.Enc}(\tilde{\rho}_i)$, and performs $(\phi_{\mathsf{anc}}^{(1)}, \cdots \phi_{\mathsf{anc}}^{(n)}) \leftarrow \mathsf{QECC.Enc}(\phi_{\mathsf{anc}})$. Next, it teleports $\tilde{\rho}_i^{(j)}, \phi_{\mathsf{anc}}^{(j)}$ back to the server j via registers $\tilde{S}_{j,i}, D_j$ respectively. If the teleportation results are $\tilde{P}_{j,i}, \tilde{P}_{j,\mathsf{anc}}$, it sets $E_{j,d} = E_j{}^{R_{j,1},\cdots,R_{j,n},N_j,T_j} \tilde{P}_{j,1}^{R_{j,1}} \cdots \tilde{P}_{j,n}^{R_{j,n}} \tilde{P}_{j,\mathsf{anc}}^{N_j}$.
- $\mathcal{H}_{3+d-k}$, where $k = d, \cdots, 1$:
  - Σ^{BoBW} step 5: For $h = d, \cdots, k$ and for every j, $\mathsf{T}^{\mathsf{hybrid}}$ resets $(N_{j,h}, T_{j,n+h})$ as a random string $\tilde{r}_{j,h} \leftarrow \mathbb{Z}_p^{\lambda+1}$, samples Clifford $V_{j,h} \leftarrow \mathscr{C}_{\ell^{\text{total}}+(n+h)\lambda}$, $\tilde{E}_{j,k-1} \leftarrow \mathscr{C}_{\ell^{\text{total}}+(n+k-1)\lambda}$ and sets $E_j = V_{j,d}^\dagger{}^{R'_{j,[n]},N'_{j,[d]},T'_{j,[n+d]}} \cdots V_{j,k}^\dagger{}^{R'_{j,[n]},N'_{j,[k]},T'_{j,[n+k]}} \tilde{E}_{j,k-1}^{R'_{j,[n]},N'_{j,[k-1]},T'_{j,[n+k-1]}}$.
  - Π^{BoBW} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ computes $\tilde{\rho}_{(k)}{}^{\tilde{R}_{[n]},\tilde{N}_{[k-1]}} \leftarrow C[k] \cdots C[d] \left(\tilde{\rho}_1^{\tilde{R}_1}, \cdots, \tilde{\rho}_n^{\tilde{R}_n}, \phi_{\mathsf{anc}}{}^{\tilde{N}}\right)$, performs $(\tilde{\rho}_{(k)}^{(1)}, \cdots, \tilde{\rho}_{(k)}^{(n)}) \leftarrow \mathsf{QECC.Enc}(\tilde{\rho}_{(k)})$, and teleports $\tilde{\rho}_{(k)}^{(j)}$ back to server j. If the teleportation result is $\tilde{P}_j$, it sets $E_{j,k} = E_j{}^{R_{j,[n]},N_{j,[k-1]},T_{j,[n+k-1]}} \tilde{P}_j^{R_{j,[n]},N_{j,[k-1]}}$.
  - Π^{BoBW} step 3 iteration h for $h = d, \cdots, k$: $\mathsf{T}^{\mathsf{hybrid}}$ sends $V_{j,h}$ to server j and receives $r_{j,h}$ in return. If $r_{j,h} \neq \tilde{r}_{j,h}$, $\mathsf{T}^{\mathsf{hybrid}}$ adds server j to the corruption list J and will not interact with this server anymore. If $|J| > t$, $\mathsf{T}^{\mathsf{hybrid}}$ publicly outputs J as malicious and aborts.
- $\mathcal{H}_{3+d}$:
  - Σ^{BoBW} step 4: $\mathsf{T}^{\mathsf{hybrid}}$ samples $(\mathsf{sk}_{j\to i}, \phi_{S,j\to i}, \sigma_{j,i}) \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Setup}}(1^\lambda, 1^{\ell_i})$, uses the content $\mathcal{R}_{j,i}$ of register $R_{j,i}$ as $\phi_{R,j\to i}$, and reassigns register $(R_{j,i}, T_{j,i})$ as $\sigma_{j,i}$.
  - Π^{BoBW} step 2: $\mathsf{T}^{\mathsf{hybrid}}$ computes the circuit C with input $(\tilde{\rho}_1^{\tilde{R}_1}, \cdots \tilde{\rho}_n^{\tilde{R}_n})$, obtains the quantum output $(\tilde{\rho}'_1, \cdots \tilde{\rho}'_n)$ of C, performs $(\tilde{\rho}_i'^{(1)}, \cdots, \tilde{\rho}_i'^{(n)}) \leftarrow \mathsf{QECC.Enc}(\tilde{\rho}'_i)$, teleports $\tilde{\rho}_i'^{(j)}$ via register $\tilde{S}_{j,i}$, and sets $\mathsf{dk}_{j,i}$ as the teleportation result $\tilde{P}_{j,i}$.
  - Π^{BoBW} step 5: $\mathsf{T}^{\mathsf{hybrid}}$ sets $V'_j = \tilde{E}_{j,0}^\dagger$.
  - Π^{BoBW} step 7: $\mathsf{T}^{\mathsf{hybrid}}$ resets $\mathsf{dk}_{j,i} = \perp$ if $\perp \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Audit}}(\mathsf{sk}_{j\to i}, \mathsf{pf}_{j,i})$.

The last hybrid is equivalent to the ideal world with the following simulator.

Simulator 2 $\mathsf{Sim}^{\mathsf{BoBW}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for BoBW-MPQC-PVIA with Trusted Setup

1. Fake setup (if client i is corrupted):
    (a) Prepare EPR pairs on registers $(S_i, \tilde{R}_i)$ with $|S_i| = |\tilde{R}_i| = \ell_i$.
    (b) Prepare EPR pairs on registers $(\tilde{S}_{j,i}, \hat{R}_{j,i})$ with $|\tilde{S}_{j,i}| = |\hat{R}_{j,i}| = \ell_i$.
    (c) Send $(S_i, \hat{R}_{j,i})$ to $\mathcal{A}$.

2. Fake setup (if server j is corrupted):
    (a) Sample $(\mathsf{sk}_{j\to i}, \phi_{S,j\to i}, {\sigma_{j,i}}^{R_{j,i},T_{j,i}}) \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Setup}}(1^\lambda, 1^{\ell_i})$ for $i \in [n]$.
    (b) Initialize registers $(N_{j,k}, T_{j,n+k})$ with random $\tilde{r}_{j,k} \leftarrow \mathbb{Z}_p^{\lambda+1}$ for $k \in [d]$.
    (c) Sample random Clifford gates $V_{j,k} \leftarrow \mathscr{C}_{\ell^{\mathrm{total}}+(n+k)\lambda}$ for $k = 0, 1, \cdots, d$.
    (d) Apply ${V_{j,d}^\dagger}^{R_{j,[n]},N_{j,[d]},T_{j,[n+d]}} \cdots {V_{j,1}^\dagger}^{R_{j,[n]},N_{j,[1]},T_{j,[n+1]}} {V_{j,0}^\dagger}^{R_{j,[n]},T_{j,[n]}}$ with resulting state σ_j.
    (e) Send $({\sigma_j}^{R_{j,1},\cdots,R_{j,n},N_j,T_j}, \phi_{S,j\to 1}, \cdots, \phi_{S,j\to n})$ to $\mathcal{A}$.

3. Input extraction: Receive P_i from $\mathcal{A}$ and extract the input ${\tilde{\rho}_i}^{\tilde{R}_i} = P_i^\dagger {\mathcal{R}_i}^{\tilde{R}_i}$.

4. Invoke the ideal functionality: Send $\tilde{\rho}_i$ to T and receive the output $\tilde{\rho}'_i$ from T.

5. Check the abort decision (if server j is corrupted):
    (a) For $k = d, \cdots, 1$, send $V_{j,k}$ to $\mathcal{A}$ and receive $r_{j,k}$ in return.
    If $r_{j,k} \neq \tilde{r}_{j,k}$, send `abort` to T in the name of server j.
    (b) Send $V_{j,0}$ to $\mathcal{A}$.
    (c) For $i \in [n]$, receive $\mathsf{pf}_{j,i}$ from $\mathcal{A}$.
    If $\perp \leftarrow \mathsf{AQA}.\widetilde{\mathsf{Audit}}(\mathsf{sk}_{j,i}, \mathsf{pf}_{j,i})$, send `abort` to T in the name of server j.

6. Output delivery: Perform $(\tilde{\rho}_i'^{(1)}, \cdots, \tilde{\rho}_i'^{(n)}) \leftarrow \mathsf{QECC.Enc}(\tilde{\rho}'_i)$, teleport $\tilde{\rho}_i'^{(j)}$ via $\tilde{S}_{j,i}$, and obtain its teleportation result $\tilde{P}_{j,i}$. Set $\mathsf{dk}_{j,i} = \tilde{P}_{j,i}$ if server j did not send `abort` in the previous step; otherwise, set $\mathsf{dk}_{j,i} = \perp$. Send $\mathsf{dk}_{j,i}$ to $\mathcal{A}$.

7. Output $\mathcal{A}$'s output.

To prove our theorem, it suffices to show indistinguishability between consecutive hybrids. In the following, $\mathsf{QECC.Enc}_j$ will represent the j-th part of the output of $\mathsf{QECC.Enc}$.

- $\mathsf{Real} = \mathcal{H}_1$: This is because $\Sigma^{\mathsf{BoBW}}, \mathsf{cMPC}$ and $\mathsf{T}^{\mathsf{hybrid}}$ are all trusted executions.

- $\mathcal{H}_1 = \mathcal{H}_2$: Consider executing the protocol until the end of step 1. Suppose the CP map A projects $(M_{[n]}, S_{[n]})$ to $|P_{[n]}\rangle$ in step 1. The state of $\mathcal{H}_1$ after protocol step 2 is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{E_j\leftarrow\mathscr{C}} \mathsf{Mixed}\left[A\left(E_j\left(\mathsf{QECC.Enc}\left(\mathcal{R}_{[n]}, \phi_{\mathsf{anc}}\right)^{R_{j,[n]},N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right] \\
=&\mathop{\mathbb{E}}_{E_{j,d}\leftarrow\mathscr{C}} \mathsf{Mixed}\left[A\left(E_{j,d}\left({P_{j,[n]}^\dagger}^{R_{j,[n]}} \mathsf{QECC.Enc}_j\left(\mathcal{R}_{[n]}, \phi_{\mathsf{anc}}\right)^{R_{j,[n]},N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right] \\
=&\mathop{\mathbb{E}}_{E_{j,d}\leftarrow\mathscr{C}} \mathsf{Mixed}\left[A\left(E_{j,d}\left(\mathsf{QECC.Enc}_j\left(P_{[n]}^\dagger \mathcal{R}_{[n]}, \phi_{\mathsf{anc}}\right)^{R_{j,[n]},N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right]
\end{aligned}$$

where the first equality is by the definition of $E_{j,d}$ and the second equality follows from $(P_{1,[n]} \otimes \cdots \otimes P_{n,[n]})^\dagger$ $\mathsf{QECC} = \mathsf{QECC}$ $P_{[n]}^\dagger$. In $\mathcal{H}_2$, we can change the order of A and the teleportation made by $\mathsf{T}^{\mathsf{hybrid}}$ because these operators act on disjoint registers. By quantum teleportation, the state of $\mathcal{H}_2$ after protocol step 2 is as follows, which is the same as in $\mathcal{H}_1$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{E_j \leftarrow \mathscr{C}} \ \mathsf{Mixed}\left[A\left(E_j\left({\tilde{P}_{j,[n]}\tilde{\rho}_{[n]}^{(j)}}^{R_{j,[n]}}, {\tilde{P}_{j,\mathsf{anc}}\phi_{\mathsf{anc}}^{(j)}}^{N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right] \\
=&\mathop{\mathbb{E}}_{E_{j,d} \leftarrow \mathscr{C}} \ \mathsf{Mixed}\left[A\left(E_{j,d}\left({\tilde{\rho}_{[n]}^{(j)}}^{R_{j,[n]}}, {\phi_{\mathsf{anc}}^{(j)}}^{N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right] \\
=&\mathop{\mathbb{E}}_{E_{j,d} \leftarrow \mathscr{C}} \ \mathsf{Mixed}\left[A\left(E_{j,d}\left(\mathsf{QECC.Enc}_j\left(P_{[n]}^\dagger \tilde{\mathcal{R}}_{[n]}, \phi_{\mathsf{anc}}\right)^{R_{j,[n]},N_j}, {|0\rangle^{\otimes(n+d)\lambda}}^{T_j}\right), \cdots\right) \otimes |E_{j,d}\rangle\right]
\end{aligned}$$

- $\mathcal{H}_{2+d-k} = \mathcal{H}_{3+d-k}$: This is similar to 6.1 except here, we allow multiple servers $j \in [n]$. Consider executing the protocol until the end of step 3 iteration k+1. Suppose the CP map A projects $(M_{j,[n]}, S_{j,[n]})$ to $|P_{j,[n]}\rangle$ in step 1 and obtains $b_d, \cdots, b_{k+1}$ in step 3 for iterations d to k+1. The partial computation result $C[k+1]\cdots C[d]\big(P_{[n]}^\dagger \tilde{\mathcal{R}}_{[n]}, \phi_{\mathsf{anc}}\big)$ is denoted as ${\tilde{\rho}_{(k)}}^{\tilde{R}_{[n]},\tilde{N}_{[k]}} = \sum_{u\in\mathbb{Z}_p} {\tilde{\rho}_{(k,u)}}^{\tilde{R}_{[n]},\tilde{N}_{[k-1]}} \otimes |u\rangle^{\tilde{N}_k}$. Let us write $\mathsf{QECC.Enc}_j(\tilde{\rho}_{(k)})$ as ${\tilde{\rho}_{(k)}^{(j)}}^{\tilde{R}_{j,[n]},\tilde{N}_{j,[k]}} = \sum_{u_j\in\mathbb{Z}_p} {\tilde{\rho}_{(k,j,u_j)}}^{\tilde{R}_{j,[n]},\tilde{N}_{j,[k-1]}} \otimes |u_j\rangle^{\tilde{N}_{j,k}}$. In $\mathcal{H}_{2+d-k}$, the state at protocol step $3(a)$ for iteration k is

$$\begin{aligned}
&\mathbb{E} \ \mathsf{Mixed}\left[A\left(E_{j,k}\left({\tilde{\rho}_{(k)}^{(j)}}^{R_{j,[n]},N_{j,[k]}}, {|0\rangle^{\otimes(n+j)\lambda}}^{T_{j,[n+k]}}\right), \cdots\right) \otimes |P'_{j,k}, c_{j,k}\rangle\right] \\
=&\mathbb{E} \ \mathsf{Mixed}\left[A\left(V_{j,k}^\dagger\left(\sum_{u_j\in\mathbb{Z}_p} E'_{j,k-1}\big(\tilde{\rho}_{(k,j,u_j)}, |0\rangle^{\otimes(n+j-1)\lambda}\big) \otimes P'_{j,k}\mathsf{CX}_{c_{j,k}}\big(|u_j\rangle, |0\rangle^{\otimes\lambda}\big)\right), \cdots\right) \otimes |P'_{j,k}, c_{j,k}\rangle\right]
\end{aligned}$$

The equality holds by the definition of $V_{j,k}$. Since the protocol gives the attacker access to $|V_{j,k}\rangle$ in step $3(a)$, we can merge $V_{j,k}^\dagger$ and A into a CP map A' that operates also on $|V_{j,k}\rangle$. Also, we can simplify $\mathsf{CX}_{c_{j,k}}(|u_j\rangle, |0\rangle^{\otimes\lambda}) = |u_j(1, c_{j,k})\rangle$. Thus, the state is equal to

$$\mathbb{E} \ \mathsf{Mixed}\left[A'\left(\sum_{u_j\in\mathbb{Z}_p} E'_{j,k-1}\big(\tilde{\rho}_{(k,j,u_j)}, |0\rangle^{\otimes(n+k-1)\lambda}\big) \otimes P'_{j,k}\,|u_j(1, c_{j,k})\rangle^{N_{j,k},T_{j,n+k}}, \cdots\right) \otimes |P'_{j,k}, c_{j,k}\rangle\right]$$

Afterwards, protocol step $3(c)$ projects registers $(N_{j,k}, T_{j,n+k})$ to $|x(P'_{j,k}) + b_{j,k}(1, c_{j,k})\rangle$ for all possible values of $b_{j,k}$. To analyze, we apply Pauli twirl with target measurement result $b_{j,k}(1, c_{j,k})$ (lemma 3.3) and use the Pauli decomposition $A' = \sum_{Q_j\in\mathscr{P}^*_{\lambda+1}} {Q_j}^{N_{j,k},T_{j,n+k}} \otimes A'_{Q_j}$.

The state after the projection (denoted $\mathsf{\Pi}_{b_{j,k}}$) of step $3(c)$ for obtaining solution $b_{j,k}$ is

$$\mathbb{E}\ \mathsf{Mixed}\left[\mathsf{\Pi}_{b_{j,k}}{}^{N_{j,k},T_{j,n+k}} A'\left(\sum_{u_j\in\mathbb{Z}_p} P'_{j,k}\left|u_j(1,c_{j,k})\right\rangle^{N_{j,k},T_{j,n+k}}\otimes E'_{j,k-1}\big(\tilde{\rho}_{(k,j,u_j)},|0\rangle^{\otimes(n+k-1)\lambda}\big),\cdots\right)\right]$$
$$=\mathbb{E}\ \sum_{u_j}\ \mathsf{Mixed}\left[\sum_{x(Q_j)=(b_{j,k}-u_j)(1,c_{j,k})} Q_j\left|\tilde{r}_{j,k}\right\rangle\otimes A'_{Q_j}\left(E'_{j,k-1}\big(\tilde{\rho}_{(k,j,u_j)},|0\rangle^{\otimes(n+k-1)\lambda}\big),\cdots\right)\right]$$

The state at step $3(c)$ conditioned that there is a solution for $b_{j,k}$ is

$$\mathop{\mathbb{E}}_{c_{j,k},\tilde{r}_{j,k},E'_{j,k\text{-}1}}\sum_{b_{j,k}}\sum_{u_j}\ \mathsf{Mixed}\left[\sum_{x(Q_j)=(b_{j,k}-u_j)(1,c_{j,k})} Q_j\left|\tilde{r}_{j,k}\right\rangle\otimes A'_{Q_j}\left(E'_{j,k-1}\big(\tilde{\rho}_{(k,j,u_j)},|0\rangle^{\otimes(n+k-1)\lambda}\big),\cdots\right)\otimes\left|b_{j,k}\right\rangle\right]$$

where the last register is the solution stored by $\mathsf{T}^{\mathsf{hybrid}}$. Observe that each $c_{j,k}$ occurs with negligible probability, so the sum of summands for $u_j\neq b_{j,k}$ have negligible trace norm when averaged over $c_{j,k}$. Hence, the above state is indistinguishable to the following state where only $u_j=b_{j,k}$ is left. Note that this state is as if the last qupit of $\tilde{\rho}^{(j)}_{(k)}$ is being measured in the computational basis and only $\mathsf{T}^{\mathsf{hybrid}}$ knows the measurement result:

$$\mathop{\mathbb{E}}_{\tilde{r}_{j,k},E'_{j,k\text{-}1}}\sum_{b_{j,k}}\ \mathsf{Mixed}\left[\sum_{x(Q_j)=0} Q_j\left|\tilde{r}_{j,k}\right\rangle\otimes A'_{Q_j}\left(E'_{j,k-1}\big(\tilde{\rho}_{(k,j,b_{j,k})},|0\rangle^{\otimes(n+k-1)\lambda}\big),\cdots\right)\otimes\left|b_{j,k}\right\rangle\right]\tag{6}$$

The state at step $3(c)$ conditioned that there is no solution for $b_{j,k}$ is indistinguishable, using a similar argument and the Clifford twirl (Lemma 3.2), to the following state:

$$\mathop{\mathbb{E}}_{\tilde{r}^*_{j,k}}\ \mathsf{Mixed}\left[\sum_{x(Q_j)\neq0}(Q_j\otimes A'_{Q_j})\left(\left|\tilde{r}^*_{j,k}\right\rangle^{R_{j,[n]},N_{j,[k]},T_{j,[n+k]}},\cdots\right)\otimes\left|1_{j\in J}\right\rangle\right]\tag{7}$$

where $1_{j\in J}$ is a flag that indicates server j is in the corruption list J. Next, we analyze the state at step $3(d)$. If the corruption list has size $|J|>t$, then J will be publicly announced and $E'_{j,k-1}$ will be discarded. By the Clifford twirl (Lemma 3.2), the residual joint state consists of a maximally mixed state and the information of J.

If $|J|\leq t$, $\mathsf{T}^{\mathsf{hybrid}}$ decodes $b_k\leftarrow\mathsf{QECC.Dec}((b_{1,k},\cdots,b_{n,k}),J)$. Since QECC tolerates t erasure errors, the values of $\{b_{j,k}\}_{j\in J}$ does not affect the QECC decoding result. The transversal measurement property of QECC shows that decoding the measurement results $b_{j,k}$ of $\tilde{N}_{j,k}$ is the same as obtaining the measurement result of $\tilde{N}_k$. The state obtained from these two procedures on registers $(R_{j,[n]},N_{j,[k-1]})$ are equal, which gives $\tilde{\rho}_{(k,j,b_{j,k})}=\mathsf{QECC.Enc}_j(\tilde{\rho}_{k,b_k})$. By plugging the equality to Equation (6), we see that the state held by server $j\notin J$ at step $3(d)$ is

$$\sum_{x(Q_j)=0} Q_j\left|\tilde{r}_{j,k}\right\rangle\otimes A'_{Q_j}\left(E'_{j,k-1}\big(\mathsf{QECC.Enc}_j\left(\tilde{\rho}_{(k,b_k)}\right),|0\rangle^{\otimes(n+k-1)\lambda}\big),\cdots\right)$$

Finally, protocol step $3(e)$ sets $E_{j,k-1} = E'_{j,k-1} G^{(j)\dagger}_{k-1}$. The state for $j \notin J$ now becomes

$$\sum_{x(Q_j)=0} Q_j \left|\tilde{r}_{j,k}\right\rangle \otimes A'_{Q_j} \left(E_{j,k-1} \big(G^{(j)}_{k-1} \mathsf{QECC.Enc}_j \left(\tilde{\rho}_{(k,b_k)} \right), |0\rangle^{\otimes(n+k-1)\lambda} \big), \cdots \right) \tag{8}$$

$$= \sum_{x(Q_j)=0} Q_j \left|\tilde{r}_{j,k}\right\rangle \otimes A'_{Q_j} \left(E_{j,k-1} \big(\mathsf{QECC.Enc}_j \left(G_{k-1}(\tilde{\rho}_{(k,b_k)}) \right), |0\rangle^{\otimes(n+k-1)\lambda} \big), \cdots \right) \tag{9}$$

$$= \sum_{x(Q_j)=0} Q_j \left|\tilde{r}_{j,k}\right\rangle \otimes A'_{Q_j} \left(E_{j,k-1} \big(\mathsf{QECC.Enc}_j \left(\tilde{\rho}_{(k-1)} \right), |0\rangle^{\otimes(n+k-1)\lambda} \big), \cdots \right) \tag{10}$$

where the first equality follows from $(G^{(1)}_{k-1}, \cdots, G^{(n)}_{k-1})$ being the fault-tolerant version of G_{k-1} and the second equality arises because $G_{k-1}(\tilde{\rho}_{(k,b_k)})$ is the result of performing the k-th iteration of computation on $\tilde{\rho}_{(k)}$. We have now described all the cases in $\mathcal{H}_{2+d-k}$.

In $\mathcal{H}_{3+d-k}$, the state after projecting to the acceptance condition $r_{j,k} = \tilde{r}_{j,k}$ is

$$\mathop{\mathbb{E}}_{\tilde{r}_{j,k}, E_j} \mathsf{Mixed} \left[\left|\tilde{r}_{j,k}\right\rangle\left\langle\tilde{r}_{j,k}\right|^{N_j, T_{j,n+k}} A' \left(\left|\tilde{r}_{j,k}\right\rangle^{N_{j,k}, T_{j,n+k}}, E_j \left(\epsilon_{(k)}, |0\rangle^{\otimes(n+k-1)\lambda} \right), \cdots \right) \right]$$

$$= \mathop{\mathbb{E}}_{\tilde{r}_{j,k}, E_j} \mathsf{Mixed} \left[\sum_{x(Q_j)=0} Q \left(\left|\tilde{r}_{j,k}\right\rangle \right) \otimes A'_{Q_j} \left(E_j \left(\epsilon_{(k)}, |0\rangle^{\otimes(n+k-1)\lambda} \right), \cdots \right) \right]$$

where $\epsilon_{(k)}$ is the half EPR pairs prepared on $R_{j,[n]}, N_{j,[k-1]}$ and the equality follows from Lemma 3.3. Next, $\mathsf{T}^{\mathsf{hybrid}}$ teleports $\mathsf{QECC.Enc}_j(\tilde{\rho}_{(k-1)})$ and fixes its teleportation Pauli to E_j which yields $E_{j,k}$. This gives the state

$$\mathop{\mathbb{E}}_{\tilde{r}_{j,k}, E_{j,k-1}} \sum_{u_j} \mathsf{Mixed} \left[\sum_{x(Q_j)=0} Q \left(\left|\tilde{r}_{j,k}\right\rangle \right) \otimes A'_{Q_j} \left(E_{j,k-1} \left(\mathsf{QECC.Enc}_j(\tilde{\rho}_{(k-1)}), |0\rangle^{\otimes(n+k-1)\lambda} \right), \cdots \right) \right] \tag{11}$$

The state after projecting to the rejection condition $r_{j,k} \neq \tilde{r}_{j,k}$ would be exactly the same as Equation (7), using a similar argument and the Clifford twirl (Lemma 3.2). Next, if $|J| > t$, then $E_{j,k-1}$ will be discarded. By Clifford twirl, the residual joint state is same as in $\mathcal{H}_{2+d-k}$. If $|J| \leq t$, the state described in Equation (11) is the same as the state described in Equation (10). Thus, $\mathcal{H}_{2+d-k}$ and $\mathcal{H}_{3+d-k}$ are indistinguishable.

- $\mathcal{H}_{2+d} = \mathcal{H}_{3+d}$: Consider executing the protocol until the end of step 6. Let $(\tilde{\rho}'_1, \cdots, \tilde{\rho}'_n)$ be the quantum output of C and $\tilde{\rho}'^{(j)}_i = \mathsf{QECC.Enc}_j(\tilde{\rho}'_i)$. The state of $\mathcal{H}_{2+d}$ at step 6 is

$$\mathop{\mathbb{E}}_{E_{j,0}, \mathsf{sk}_{j\to i}} \mathsf{Mixed} \left[A \left(E_{j,0}{}^{R_{j,[n]}, T_{j,[n]}} \left(\tilde{\rho}'^{(j)}_i{}^{R_{j,i}}, |0\rangle^{\otimes\lambda\, T_{j,i}} \right), |V'_j\rangle, \phi_{S,j\to i}, \phi_{R,j\to i}, \cdots \right) \otimes |\mathsf{sk}_{j\to i}\rangle \right]$$

$$= \mathop{\mathbb{E}}_{V'_j, \mathsf{sk}_{j\to i}} \mathsf{Mixed} \left[A \left(V'^{\dagger}_j{}^{R_{j,[n]}, T_{j,[n]}} \left(\mathsf{AQA.EncG}(\mathsf{sk}_{j\to i}) \left(\tilde{\rho}'^{(j)}_i, |0\rangle^{\otimes\lambda} \right)^{R_i, T_i} \right), |V'\rangle, \phi_{S,i}, \phi_{R,i}, \cdots \right) \otimes |\mathsf{sk}_{j\to i}\rangle \right]$$

$$= \mathop{\mathbb{E}}_{V'_j, \mathsf{sk}_{j\to i}} \mathsf{Mixed} \left[A \left(V'^{\dagger}_j{}^{R_{j,[n]}, T_{j,[n]}} \left(\mathsf{AQA.Enc} \left(\mathsf{sk}_{j\to i}, \tilde{\rho}'^{(j)}_i \right)^{R_i, T_i} \right), |V'\rangle, \phi_{S,i}, \phi_{R,i}, \cdots \right) \otimes |\mathsf{sk}_{j\to i}\rangle \right]$$

where the first equality follows from the definition of V' in $\mathcal{H}_{2+d}$ and the second follows from the definition of $\mathsf{AQA.Enc}$. After protocol step 7, the state of $\mathcal{H}_{2+d}$ can be generated under the process $(\mathsf{AQA.Setup}, \mathsf{AQA.Enc}, A', \mathsf{Audit})$ where A' is a CP map that merges A and $V'^{\dagger}$. On the other hand, the state of $\mathcal{H}_{3+d}$ at step 6 is

$$\begin{aligned}
&\underset{\tilde{E}_0,\mathsf{sk}_{j\to i},\tilde{P}_{j,i}}{\mathbb{E}} \ \mathsf{Mixed}\left[A\left(\tilde{E}_0^{R_{j,[n]},T_{j,[n]}}\left({\sigma_{j,i}}^{R_{j,i},T_{j,i}}\right), |V_j'\rangle, \phi_{S,j\to i}, \tilde{P}_{j,i}(\tilde{\rho}_i'^{(j)}), \cdots\right) \otimes |\tilde{P}_{j,i}\rangle \otimes |\mathsf{sk}_{j\to i}\rangle\right] \\
= &\underset{V_j',\mathsf{sk}_{j\to i},\mathsf{dk}_{j,i}}{\mathbb{E}} \ \mathsf{Mixed}\left[A\left({V_j'^{\dagger}}^{R_{j,[n]},T_{j,[n]}}\left({\sigma_{j,i}}^{R_{j,i},T_{j,i}}\right), |V_j'\rangle, \phi_{S,j\to i}, \mathsf{dk}_{j,i}(\tilde{\rho}_i'^{(j)}), \cdots\right) \otimes |\mathsf{dk}_{j,i}\rangle \otimes |\mathsf{sk}_{j\to i}\rangle\right]
\end{aligned}$$

where the equality follows from the definition of V_j' and $\mathsf{dk}_{j,i}$ in $\mathcal{H}_{3+d}$. Here, $\mathsf{sk}_{j\to i}, \phi_{S,j\ toi}, \sigma_{j,i}$ are generated from $\mathsf{AQA.}\widetilde{\mathsf{Setup}}$, whereas $\mathsf{dk}_{j,i}$ and the encoding of $\tilde{\rho}_i'^{(j)}$ can be generated from $\mathsf{AQA.}\widetilde{\mathsf{Enc}}$. After protocol step 7, the state of $\mathcal{H}_{3+d}$ can be generated under the process $(\mathsf{AQA.}\widetilde{\mathsf{Setup}}, \mathsf{AQA.}\widetilde{\mathsf{Enc}}, A', \widetilde{\mathsf{Audit}})$. By the receiver security of AQA (Theorem 5.2), these two states are statistically indistinguishable.

- $\mathcal{H}_{3+d} = \mathsf{Ideal}$: Except for the computation of C, the execution of $\mathsf{T}^{\mathsf{hybrid}}$ in $\mathcal{H}_{3+d}$ consists of n independent parts, each interacting with only one party. Thus, we can collect the parts that interact with the corrupted parties as a simulator Sim, who also interacts with the ideal functionality of MPQC to compute C. By quantum teleportation, the completeness and security of our AQA, and the recoverability of QECC, the honest parties and their corresponding parts of $\mathsf{T}^{\mathsf{hybrid}}$ indeed send the honest inputs to the ideal functionality, never send `abort` to the ideal functionality, and always output the honest outputs. Hence, $\mathcal{H}_{3+d}$ is identical to Ideal.

□

8 BoBW-MPQC-PVIA without Trusted Setup

So far, we have demonstrated an BoBW-MPQC-PVIA protocol in the preprocessing model with a trusted setup. Now, we will illustrate how to instantiate the preprocessing phase *without* a trusted setup. Our approach involves using MPQC-SWIA e.g., [ACC+21] to implement the preprocessing phase. Note that [ACC+21] requires a post-quantum fully homomorphic encryption assumption to achieve the security; however, we do not need this assumption when we want to compute circuits that have no inputs and apply only Clifford gates to $|0\rangle$'s and $|\mathsf{T}\rangle$'s. We defer the explicit MPQC-SWIA construction to Appendix C. We reformulate MPQC-SWIA as the following lemma.

Lemma 8.1 (MPQC-SWIA)**.** *There is a multi-party quantum computation* Π^{SWIA} *secure with identifiable abort in the MPC-hybrid model that computes quantum circuit* C*, given that* C *takes no inputs and only applies Clifford to ancillary* $|0\rangle$ *and* $|\mathsf{T}\rangle$ *states. That is, for every non-uniform* (QPT) *adversary* $\mathcal{A}$ *corrupting party set* I*, there is a non-uniform* (QPT) *adversary* $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{SWIA}}$ *corrupting* I*, such that*

$$\{\mathsf{Real}_{\mathcal{A}(\rho_{\mathsf{aux}})}^{\Pi^{\mathsf{SWIA}}}(1^\lambda, C)\} \approx \{\mathsf{Ideal}_{\mathsf{Sim}_{\mathcal{A}(\rho_{\mathsf{aux}})}^{\mathsf{SWIA}}}^{\mathsf{SWIA}}(1^\lambda, C)\}$$

$\mathsf{Ideal}^{\mathsf{SWIA}}$: **Multi-party Quantum Computation Secure with Identifiable Abort**

Common input:

The security parameter 1^λ and a quantum circuit C with no inputs.

Input:

$\mathcal{A}_I$ holds input ρ_{aux} and controls parties in I.

Execution of $\mathsf{T}^{\mathsf{SWIA}}$:

$\mathsf{T}^{\mathsf{SWIA}}$ computes $(\rho'_1, \rho'_2, \cdots, \rho'_n, r'_{\mathsf{cMPC}}) \leftarrow C$, where r'_{cMPC} is classical.
$\mathsf{T}^{\mathsf{SWIA}}$ sends ρ'_i to all $\mathsf{P}_i \in I$. Every $\mathsf{P}_i \in I$ can send `abort` message to $\mathsf{T}^{\mathsf{SWIA}}$.
Let $I_{\texttt{abort}}$ be the set of parties who indeed send the `abort` message.
If $I_{\texttt{abort}}$ is non-empty, $\mathsf{T}^{\mathsf{SWIA}}$ sends the partition $\{I_{\texttt{abort}}, \mathsf{P}_{[n]} \backslash I_{\texttt{abort}}\}$ to all parties.
Otherwise, $\mathsf{T}^{\mathsf{SWIA}}$ sends ρ'_i to all $\mathsf{P}_i \notin I$ and sends r'_{cMPC} to cMPC.

Output:

Honest parties output whatever output received from $\mathsf{T}^{\mathsf{SWIA}}$.
The adversary $\mathcal{A}_I$ outputs a function of his view.

We say an execution succeeds if every party receives their part of the circuit output. Whenever the execution fails, all parties get to know how they have been partitioned into two groups. An honest party can infer that the group he does not belong to (i.e., $I_{\texttt{abort}}$) is the set of malicious parties who interfered the computation.

In contrast to [ACC+21] where the partition information serves the purpose of SWIA, we utilize the partition information to design a preprocessing procedure that *always* succeeds. Such a preprocessing procedure will be suitable for replacing the setup of MPQC-PVIA.

8.1 Protocol

First, we let the parties run MPQC-SWIA over the setup circuit Σ^{BoBW} that the trusted setup is supposed to run. If the MPQC-SWIA fails, the parties will be divided into two groups. Each group will then ignore other groups and run MPQC-SWIA independently. By iterating this process, each party will eventually find a group in which the MPQC-SWIA succeeds and receive an output of the setup circuit. This constitutes our preprocessing procedure. Note that we do not publicly identify anyone during this stage.

After obtaining the output of the setup circuit, every party can run the protocol Π^{BoBW} within their group to obtain their MPQC output. It is reasonable for each group to operate independently, as they view other groups as untrustworthy. Parties within a group can set the inputs of the parties outside the group as some default inputs such as $|0\rangle$. Moreover, MPQC-SWIA guarantees that the honest parties are always in the same group, so they will jointly compute their outputs. Our approach circumvents the false accusation problem encountered in [ACC+21] because the parties in our protocol no longer accuse between groups. Instead, each group runs its own Π^{BoBW}, which only aborts dishonest members. Below is the formulation of our protocol.

Protocol 3 Π^{MPQC} for BoBW-MPQC-PVIA

Input:

1. Everyone holds the threshold t and the circuit description C.
2. Party P_i holds private input $\rho_i \in \mathcal{D}^{\ell_i}$.

Protocol:

1. Set $G = \{\{\mathsf{P}_1, \cdots, \mathsf{P}_n\}\}$ as the initial partition of parties (i.e., no partition) and mark the set $\{\mathsf{P}_1, \cdots, \mathsf{P}_n\} \in G$ as unfinished.
2. Repeat the following as long as G contains a set S that is marked as unfinished:
   (a) The parties in S run Π^{SWIA} over the setup circuit $\Sigma^{\mathsf{BoBW}}(t, C_S)$, where C_S is the circuit that prepares default inputs (e.g., $|0\rangle$) for parties not in S and runs C.
   (b) If Π^{SWIA} succeeds or if $|S| = 1$, parties in S has obtained the output of the setup circuit. In this case, mark S as finished.
   (c) Otherwise, Π^{SWIA} instructs to partition S into $\{S_0, S_1\}$ of S. In this case, replace S with S_0 and S_1 in G, and mark both S_0, S_1 as unfinished.
3. Run $\Pi^{\mathsf{BoBW}}(t, C_S)$ within every set $S \in G$.
4. Let J be the union of all corruption lists output by all executions of Π^{BoBW}. We note that every corruption list is public and only contains the parties that participate.

Output:

1. If $|J| > t$, every party outputs J as malicious and aborts.
2. Otherwise, P_i outputs the result obtained from his execution of Π^{BoBW}.

8.2 Security

Theorem 8.2. Π^{MPQC} *is a best-of-both-worlds multi-party quantum computation secure with publicly verifiable identifiable abort of threshold t in the MPC-hybrid model.*

Proof. We would like to show that the real world and the ideal world are indistinguishable using the following simulator. We regard $\mathcal{A}$ as a stateful adversary throughout.

Simulator 3 $\mathsf{Sim}^{\mathsf{MPQC}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for BoBW-MPQC-PVIA

1. Set $S = \{\mathsf{P}_1, \cdots, \mathsf{P}_n\}$ as the initial group that contains all the honest parties.
2. Repeat the following as long as the inner $\mathsf{Ideal}^{\mathsf{SWIA}}$ fails:
   (a) Run a simulated $\mathsf{Ideal}^{\mathsf{SWIA}}$ that computes the first part of $\mathsf{Sim}^{\mathsf{BoBW}}_{\mathcal{A}}(C_S)$, which is a circuit that resembles the setup circuit $\Sigma^{\mathsf{BoBW}}(C_S)$ but prepares different states.

(b) Run $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{SWIA}}$ who interacts with the simulated $\mathsf{Ideal}^{\mathsf{SWIA}}$.

(c) Upon failure, let $I_{\texttt{abort}}$ be the set of parties instructed by $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{SWIA}}$ to send `abort`.

(d) Run $\mathcal{A}$ to complete the steps in the protocol that involve only the malicious group $I_{\texttt{abort}}$.

(e) Set S as the updated group $S \backslash I_{\texttt{abort}}$ that contains all the honest parties.

3. For every corrupted party not in S, send their default input to the ideal functionality.

4. Run the remaining part of $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{BoBW}}(C_S)$, which interacts with the ideal functionality.

5. Let J be the union of all corrupted lists output by all executions of Π^{BoBW}. Make all parties in J send `abort` to the ideal functionality.

6. Output the output of $\mathcal{A}$.

To prove indistinguishability, we introduce the following hybrids. We describe each hybrid in terms of the changes made to the previous hybrid, starting with $\mathcal{H}_0 = \mathsf{Real}$.

- $\mathcal{H}_i$ for $i = 1, \cdots, n$: Replace the i-th real execution of Π^{SWIA} with an ideal execution of $\mathsf{Ideal}^{\mathsf{SWIA}}$ that computes the corresponding setup circuit $\Sigma^{\mathsf{BoBW}}(C_S)$.

- $\mathcal{H}_{n+i}$ for $i = 1, \cdots, n$: If the i-th execution of $\mathsf{Ideal}^{\mathsf{SWIA}}$ will involve all the honest parties, let the ideal execution compute the first part of the simulator $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{BoBW}}(C_S)$ instead. When this execution succeeds, replace the corresponding real execution of Π^{BoBW} in protocol step 3 with an ideal execution of $\mathsf{Ideal}^{\mathsf{MPQC}}$ that involves only parties in S and interacts with the remaining part of the simulator $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{BoBW}}(C_S)$. Parties in S still follows the output procedure of the protocol instead of directly outputting what they get from $\mathsf{Ideal}^{\mathsf{MPQC}}$.

- $\mathcal{H}_{2n+1}$: Replace the ideal execution of $\mathsf{Ideal}^{\mathsf{MPQC}}(C_S)$ that involves only a subset S of parties with an ideal execution of $\mathsf{Ideal}^{\mathsf{MPQC}}(C)$ that involves all parties, where the parties not in S always send default inputs. For every corruption list output by Π^{BoBW}, let the malicious parties in the corruption list send `abort` to $\mathsf{Ideal}^{\mathsf{MPQC}}$. Parties in S directly outputs what they get from $\mathsf{Ideal}^{\mathsf{MPQC}}$.

Since the sub-protocols Π^{SWIA} are executed sequentially among the protocol, lemma 8.1 implies that $\mathcal{H}_{i-1}$ and $\mathcal{H}_i$ are indistinguishable for $i = 1, \cdots, n$. By theorem 7.1, the hybrids $\mathcal{H}_{n+i-1}$ and $\mathcal{H}_{n+i}$ are indistinguishable for $i = 1, \cdots, n$. We have $\mathcal{H}_{2n} = \mathcal{H}_{2n+1}$ by the definition of C_S and the observation that both hybrids make the same abort decisions. Moreover, if we merge all the simulators and all the ideal functionalities except for $\mathsf{Ideal}^{\mathsf{MPQC}}$ in the last hybrid, we would get simulator 3. Thus, the last hybrid has the same output as running the ideal world with simulator 3. We therefore conclude that the real and ideal worlds are indistinguishable.

□

9 Discussion and Open Questions

Alternative Forms of AQA In our paper, we construct a Clifford-form AQA. It is interesting to consider Trap-code or other forms of AQA. This opens up the feasibility of constructing MPQC or other protocols using different forms, whereas existing MPQC works in the dishonest majority setting follow only a rule of thumb along the line of the Clifford authentication code [DNS12, DGJ+20, ACC+21, BCKM21].

Applications of AQA Our AQA primitive paves the way for developing cryptographic protocols or primitives that offer public verifiability. For example, AQA may help construct publicly verifiable quantum fully homomorphic encryption (pvQFHE). However, one may need to seek a public-key version of AQA to reduce the extra round induced by the audit.

Constant-Round MPQC-PVIA. Upon close examination of MPQC-PVIA, it becomes apparent that by incorporating a quantum garbled circuit from [BCKM21] and a trusted setup, a constant-round MPQC-PVIA protocol can be achieved without difficulty. In fact, upon careful analysis, it is possible to condense the protocol to just three rounds. However, this requires a trusted setup, which is unfavorable in most cryptographic scenarios. Thus, constructing an instantiable constant-round setup that removes the trusted setup assumption would be a good direction.

Acknowledgement

The authors would like to thank Andrea Coladangelo for useful discussions. This research is supported by NSF CAREER award 2141536 and supported by NSTC QC project under Grant no. NSTC 111-2119-M-001-004.

References

[ABDR04] Andris Ambainis, Harry Buhrman, Yevgeniy Dodis, and Hein Rohrig. Multiparty quantum coin flipping. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 250–259. IEEE, 2004.

[ABO97] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 176–188, 1997.

[ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint arXiv:1704.04487*, 2017.

[ACC+21] Bar Alon, Hao Chung, Kai-Min Chung, Mi-Ying Huang, Yi Lee, and Yu-Ching Shen. Round efficient secure multiparty quantum computation with identifiable abort. In *Annual International Cryptology Conference*, pages 436–466. Springer, 2021.

[BCG$^+$02] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 449–458. IEEE, 2002.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In *Annual International Cryptology Conference*, pages 406–435. Springer, 2021.

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for qma. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40. IEEE, 2016.

[BK05] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.

[BLOO11] Amos Beimel, Yehuda Lindell, Eran Omri, and Ilan Orlov. 1/p-secure multiparty computation without honest majority and the best of both worlds. In *Annual Cryptology Conference*, pages 277–296. Springer, 2011.

[BOCG$^+$06] Michael Ben-Or, Claude Crepeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006.

[BOSSV20] Carsten Baum, Emmanuela Orsini, Peter Scholl, and Eduardo Soria-Vazquez. Efficient constant-round mpc with identifiable abort and public verifiability. In *Annual International Cryptology Conference*, pages 562–592. Springer, 2020.

[CAB12] Earl T Campbell, Hussain Anwar, and Dan E Browne. Magic-state distillation in all prime dimensions using quantum reed-muller codes. *Physical Review X*, 2(4):041021, 2012.

[CGS02] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC '02*. ACM Press, 2002.

[Cha05] Hoi Fung Chau. Unconditionally secure key distribution in higher dimensions by depolarization. *IEEE Transactions on Information Theory*, 51(4):1451–1468, 2005.

[Cla06] Sean Clark. Valence bond solid formalism for d-level one-way quantum computation. *Journal of Physics A: Mathematical and General*, 39(11):2701, 2006.

[CLLW22] Kai-Min Chung, Yi Lee, Han-Hsuan Lin, and Xiaodi Wu. Constant-round blind classical verification of quantum sampling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 707–736. Springer, 2022.

[DGJ$^+$20] Yfke Dulek, Alex B Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multi-party quantum computation with a dishonest majority. *Advances in Cryptology - EUROCRYPT 2020.*, 2020.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Annual Cryptology Conference*, pages 794–811. Springer, 2012.

[GJMZ23] Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. Commitments to quantum states. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1579–1588, 2023.

[GYW23] Tianren Gu, Xiao Yuan, and Bujiao Wu. Efficient measurement schemes for bosonic systems. *Quantum Science and Technology*, 8(4):045008, 2023.

[IKK$^+$11] Yuval Ishai, Jonathan Katz, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. On achieving the "best of both worlds" in secure multiparty computation. *SIAM journal on computing*, 40(1):122–141, 2011.

[IOZ14] Yuval Ishai, Rafail Ostrovsky, and Vassilis Zikas. Secure multi-party computation with identifiable abort. In *Advances in Cryptology – CRYPTO 2014*, pages 369–386. Springer Berlin Heidelberg, 2014.

[Kat07] Jonathan Katz. On achieving the" best of both worlds" in secure multiparty computation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 11–20, 2007.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[MGW87] Silvio Micali, Oded Goldreich, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*, pages 218–229. ACM, 1987.

[RBO89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85, 1989.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–505. Springer, 2010.

[VDB21] Ewout Van Den Berg. A simple method for sampling random clifford operators. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 54–59. IEEE, 2021.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, 1986.

A Proof of Lemmas

Lemma A.1. *Let $u \neq u' \in \mathbb{Z}_p^n$ and $Q, Q' \in \mathscr{P}_n$ such that $x(Q) + u = x(Q') + u'$. Then*

$$\mathop{\mathbb{E}}_{P \leftarrow \mathscr{P}_n} \left[QP \left| u \right\rangle \left\langle u' \right| P^\dagger Q'^\dagger \right] = 0$$

Proof. Let $v = x(Q) + u = x(Q') + u'$. Then $\mathbb{E}_{P \leftarrow \mathscr{P}_n} \left[QP \left| u \right\rangle \left\langle u' \right| P^\dagger Q'^\dagger \right]$ is equal to

$$\begin{aligned}
& \mathop{\mathbb{E}}_{x(P), z(P) \leftarrow \mathbb{Z}_p^n} \left[\mathsf{X}^{x(Q)} \mathsf{Z}^{z(Q)} \mathsf{X}^{x(P)} \mathsf{Z}^{z(P)} \mathsf{X}^{u} \left| 0 \right\rangle\!\left\langle 0 \right| \mathsf{X}^{-u'} \mathsf{Z}^{-z(P)} \mathsf{X}^{-x(P)} \mathsf{Z}^{-z(Q')} \mathsf{X}^{-x(Q')} \right] \\
= & \mathop{\mathbb{E}}_{x(P), z(P) \leftarrow \mathbb{Z}_p^n} \left[\omega^{(u-u')z(P) + uz(Q) - u'z(Q') + x(P)(z(Q) - z(Q'))} \left| x(Q) + x(P) + u \right\rangle \left\langle x(Q') + x(P) + u' \right| \right] \\
= & \mathop{\mathbb{E}}_{z(P) \leftarrow \mathbb{Z}_p^n} \left[\omega^{(u-u')z(P)} \mathop{\mathbb{E}}_{x(P) \leftarrow \mathbb{Z}_p^n} \left[\omega^{uz(Q) - u'z(Q') + x(P)(z(Q) - z(Q'))} \left| x(P) + v \right\rangle \left\langle x(P) + v \right| \right] \right] = 0
\end{aligned}$$

where the last equation follows from $u \neq u'$ and averaging over $z(P)$. □

Lemma A.2 (Lemma 3.3)**.** *Let $\left| \phi \right\rangle^{M,N} = \sum_{u \in \mathbb{Z}_p^n} \left| u \right\rangle^{M} \otimes \left| \phi_u \right\rangle^{N}$ be a state and $v \in \mathbb{Z}_p^n$ be the target measurement result. For any attack $A^{M,N} = \sum_{Q \in \mathscr{P}_n^*} \left(Q^{M} \otimes A_Q{}^{N} \right)$ applied on the QOTP-protected state, it holds that*

$$\begin{aligned}
& \mathop{\mathbb{E}}_{P \leftarrow \mathscr{P}_n} \mathsf{Mixed} \left[\left| v + x(P) \right\rangle\!\left\langle v + x(P) \right|^{M} A^{M,N} P^{M} \left| \phi \right\rangle^{M,N} \right] \\
= & \mathop{\mathbb{E}}_{r \leftarrow \mathbb{Z}_p^n} \sum_{u} \mathsf{Mixed} \left[\left(\sum_{x(Q) = v - u} Q \otimes A_Q \right) \left(\left| r \right\rangle \otimes \left| \phi_u \right\rangle \right) \right]
\end{aligned}$$

Proof.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[|v+x(P)\rangle\langle v+x(P)|^M A^{M,N} P^M |\phi\rangle^{M,N}\right] \\
=&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[|v+x(P)\rangle\langle v+x(P)|^M \sum_{Q\in\mathscr{P}_n^*}\sum_{u\in\mathbb{Z}_p^n} \left(Q^M \otimes {A_Q}^N\right) P^M \left(|u\rangle^M \otimes |\phi_u\rangle^N\right)\right] \\
=&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[|v+x(P)\rangle\langle v+x(P)|^M \sum_{Q\in\mathscr{P}_n^*}\sum_{u\in\mathbb{Z}_p^n} \left(QP\,|u\rangle^M \otimes A_Q\,|\phi_u\rangle^N\right)\right] \\
=&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \mathsf{Mixed}\left[\sum_{u\in\mathbb{Z}_p^n}\sum_{x(Q)=v-u} QP\,|u\rangle^M \otimes A_Q\,|\phi_u\rangle^N\right] \\
=&\mathop{\mathbb{E}}_{P\leftarrow\mathscr{P}_n} \sum_{u\in\mathbb{Z}_p^n} \mathsf{Mixed}\left[\sum_{x(Q)=v-u} QP\,|u\rangle^M \otimes A_Q\,|\phi_u\rangle^N\right] \\
=&\mathop{\mathbb{E}}_{r\leftarrow\mathbb{Z}_p^n} \sum_{u\in\mathbb{Z}_p^n} \mathsf{Mixed}\left[\sum_{x(Q)=v-u} Q\,|r\rangle^M \otimes A_Q\,|\phi_u\rangle^N\right]
\end{aligned}$$

The fourth line holds because $v + x(P) = x(Q) + x(P) + u$ if and only if $x(Q) = v - u$. The fifth line follows from expanding $\mathsf{Mixed}\,[\cdot]$ and eliminating the cross terms for $u, u' \in \mathbb{Z}_p^n$ using Lemma A.1. The last line follows from the Pauli twirl (Lemma 3.2). □

Lemma A.3 (Lemma 3.9)**.** *Let (ψ,τ) be a purified state independent of (e_S, e_R). Then*

$$\left(\mathsf{TP.Send}(\psi^M, {e_S}^S), {e_R}^R, \tau^N\right) = \frac{1}{p^n}\sum_{x,z\in\mathbb{Z}_p^n} |z\rangle^M \otimes |x\rangle^S \otimes \left(\left(\mathsf{X}^x\mathsf{Z}^z\right)\psi^R, \tau^N\right)$$

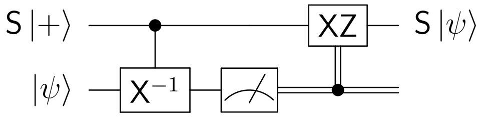
Proof. Let $(\phi,\tau) = \sum_u |u\rangle \otimes |\tau_u\rangle$. Then

$$\begin{aligned}
&\mathsf{TP}^{M,S}(\psi^M, {e_S}^S, {e_R}^R, \tau^N) \\
=&\mathsf{H}^M {\mathsf{CX}^\dagger}^{M,S}(\psi^M, {e_S}^S, {e_R}^R, \tau^N) \\
=&\mathsf{H}^M {\mathsf{CX}^\dagger}^{M,S}\left(\frac{1}{\sqrt{p^n}}\sum_{u,v} |u^M, v^S, v^R\rangle \otimes {\tau_u}^N\right) \\
=&\mathsf{H}^M\left(\frac{1}{\sqrt{p^n}}\sum_{u,v} |u^M, v-u^S, v^R\rangle \otimes {\tau_u}^N\right) \\
=&\frac{1}{p^n}\sum_{u,v,z} \omega^{\langle z,u\rangle}\,|z^M, v-u^S, v^R\rangle \otimes {\tau_u}^N \\
=&\frac{1}{p^n}\sum_{u,x,z} \omega^{\langle z,u\rangle}\,|z^M, x^S, x+u^R\rangle \otimes {\tau_u}^N \\
=&\frac{1}{p^n}\sum_{u,x,z} \mathsf{X}^x{\mathsf{Z}^z}^R\,|z^M, x^S, u^R\rangle \otimes {\tau_u}^N \\
=&\frac{1}{p^n}\sum_{x,z} |z\rangle^M \otimes |x\rangle^S \otimes \left((\mathsf{X}^x\mathsf{Z}^z)\psi^R, \tau^N\right)
\end{aligned}$$

□

B State Injection

Phase state $|\mathsf{S}\rangle = \mathsf{S}\,|+\rangle$
We can apply the phase gate S on a state $|\psi\rangle$ using CX^{-1}, X, Z gates and $|\mathsf{S}\rangle$ state as follows:



Proof. Write $\mathsf{S}\,|+\rangle = \frac{1}{\sqrt{p}}\sum_{j=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\,|j\rangle$ and $|\psi\rangle = \sum_{i=0}^{p-1}\alpha_i\,|i\rangle$. After applying CX^{-1}, we get

$$\begin{aligned}
\mathsf{CX}^{-1}(\mathsf{S}\,|+\rangle\,, |\psi\rangle) &= \mathsf{CX}^{-1}\left(\sum_{j=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\,|j\rangle\,, \sum_{i=0}^{p-1}\alpha_i\,|i\rangle\right)\\
&= \sum_{j=0}^{p-1}\sum_{i=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\alpha_i\,|j\rangle\otimes|i-j\rangle\\
&\overset{k=i-j}{=} \sum_{j=0}^{p-1}\sum_{k=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\alpha_{j+k}\,|j\rangle\otimes|k\rangle
\end{aligned}$$

where the subscript of α is taken modulo p. Then we measure the second register and obtain a measurement result k'. We now show by induction that applying $(\mathsf{XZ})^{k'}$ to the first register produces $\mathsf{S}\,|\psi\rangle$.

- If $k' = 0$, the state in the first register is $\sum_{j=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\alpha_j\,|j\rangle = \mathsf{S}\,|\psi\rangle$.
- If $k' > 0$, the state in the first register is $\sum_{j=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\alpha_{j+k'}\,|j\rangle$. Applying XZ to the state gives

  $$\sum_{j=0}^{p-1}\omega^j\omega^{\frac{j(j-1)}{2}}\alpha_{j+k'}\,|j+1\rangle = \sum_{j=0}^{p-1}\omega^{\frac{(j+1)j}{2}}\alpha_{j+k'}\,|j+1\rangle = \sum_{j=1}^{p}\omega^{\frac{j(j-1)}{2}}\alpha_{j+k'-1}\,|j\rangle$$

  Since $\omega^d = 1$, the term with $j = p$ in the last expression is equal to the term corresponding to $j = 0$. Thus, the state is equivalent to $\sum_{j=0}^{p-1}\omega^{\frac{j(j-1)}{2}}\alpha_{j+k'-1}\,|j\rangle$, which is what we would obtain if the measurement result were $k' - 1$. Hence, applying XZ and the above argument k' times would result in $\mathsf{S}\,|\psi\rangle$.

□

C MPQC-SWIA

This section explains how we obtain the MPQC-SWIA protocol we need by utilizing two primitives proposed in [ACC+21] to perform quantum computation that involves no inputs under fewer assumptions. In particular, our MPQC-SWIA protocol only assumes

cMPC whereas [ACC+21] assumes cMPC and a post-quantum fully homomorphic encryption (cFHE). First, we recall some primitives of [ACC+21].

- Authentication Routing (AR): The protocol takes as input the identity of two parties, one called the sender and the other called the receiver. The sender takes $k > n^2$ quantum packets $(\sigma_1, \cdots, \sigma_k)$ as input. At the end of the protocol, either the receiver obtains at least $k - n^2$ of the packets with known indices, or the protocol aborts and a partition of parties is known to everyone.The protocol directly generalizes to the qupit version using Clifford group for qupits and replacing $\mathrm{GL}(2, n)$ with $\mathrm{GL}(p, n)$.

- T-Magic State Preparation (MSP): The protocol takes parameters $1^\lambda, 1^n, 1^m$ as input. At the end of the protocol, either P_1 receives $E(|\mathsf{T}\rangle^{\otimes n}, |0\rangle^{\otimes(m+n\lambda)})$ and cMPC receives E for a random Clifford E, or the protocol aborts and a partition of parties is known to everyone. The protocol directly generalizes to the qupit version using Clifford group for qupits and the magic state distillation of [CAB12].

Protocol 4 Our protocol Π^{SWIA} for MPQC-SWIA

Common input: The parties hold the security parameter 1^λ and the circuit description, which samples a Clifford G from a distribution $\mathbb{G}$ and applies it to the state $|\mathsf{T}\rangle^{\otimes n} \otimes |0\rangle^{\otimes(m+n\lambda)}$.

1. Compute the following n^3 times. In the k-th iteration:
   - Call MSP so that P_1 receives a state $E_k(|\mathsf{T}\rangle^{\otimes n}, |0\rangle^{\otimes(m+n\lambda)})$ and cMPC receives E_k.
   - If MSP aborts, everyone receives a partition of parties and aborts.
   - cMPC samples $G_k \leftarrow \mathbb{G}$ and sends the Clifford gate $V_k = G_k E_k^\dagger$ to P_1.
   - P_1 applies V_k to $E_k(|\mathsf{T}\rangle^{\otimes n}, |0\rangle^{\otimes(m+n\lambda)})$ and obtains $(\sigma_{1,k}, \cdots, \sigma_{n,k})$.
2. P_1 sends the states $\{\sigma_{i,k}\}_{k\in[n^3]}$ to P_i in the order of $i = 2, \cdots, n$:
   - P_1 runs AR to send $\{\sigma_{i,k}\}_{k\in K_{i-1}}$ to P_i where we set $K_1 = [n^3]$.
   - If AR aborts, everyone receives a partition of parties and aborts.
   - cMPC sets K_i as the indices of the packets that are successfully sent through AR.
3. cMPC sends the smallest index k^* in K_n to everyone.
4. Party P_i outputs σ_{i,k^*} and cMPC outputs G_{k^*}.

Initially, there are n^3 states prepared. Each execution of AR will either abort or drop up to n^2 packets. If P_1 sends states to $\mathsf{P}_2, \cdots, \mathsf{P}_n$ with no abort, there will be no more than $n^2(n-1) < n^3$ packet drops. Hence, the set K_n will retain at least one element, which points to an iteration where all parties receive the circuit output successfully.