From Auditable Quantum Authentication to Best-of-Both-Worlds Multiparty Quantum Computation with Public Verifiable Identifiable Abort

Mi-Ying (Miryam) Huang[1] and Er-Cheng Tang[2]

[1]Department of Computer Science, University of Southern California, USA
[2]Institute of Information Science, Academia Sinica, Taiwan

November 1st, 2022

Abstract

We construct the first secure multiparty quantum computation with public verifiable identifiable abort (MPQC-PVIA) protocol, where PVIA security enables outside observers with only classical computational power to agree on the identity of a malicious party in case of an abort. Moreover, our MPQC is the first quantum setting to provide Best-of-Both-Worlds (BoBW) security, which attains full security with an honest majority and is secure with abort if the majority is dishonest. At the heart of our construction is a generic transformation called *Auditable Quantum Authentication* (AQA) that publicly identifies the malicious sender with overwhelming probability. Our approach comes with several advantages over the traditional way of building MPQC protocols. First, instead of following the Clifford code paradigm, our protocol can be based on a variety of authentication codes. Second, the online phase of our MPQC requires only classical communications. Third, our construction can achieve distributed computation via a carefully crafted protocol design, which can be adjusted to an MPQC that conditionally guarantees output delivery.

Contents

1 Introduction **3**
1.1 Our results . . . 4
1.2 Open Problems . . . 5

2 Technical overview **5**
2.1 Quantum Background . . . 6
2.2 Public Verifiability is Hard to Achieve on MPQC . . . 7
2.3 Our Solution: Auditable Quantum Authentication (AQA) . . . 7
2.4 From AQA to MPQC-PVIA . . . 9
2.5 Difficulty of Best-of-Both-Worlds Security . . . 10

3 Preliminary **11**
3.1 Quantum Teleportation . . . 12
3.2 Quantum One-Time Pad . . . 12
3.3 Twirling . . . 13
3.4 Quantum Error-Correcting Codes . . . 13
3.5 Quantum Authentication Code . . . 14
3.5.1 Clifford Code . . . 14
3.5.2 Trap Authentication Code . . . 14

4 Model and Definition **15**
4.1 Best-of-Both-Worlds Multi-Party Quantum Computation Secure with Public Verifiable Identifiable Abort . . . 15
4.2 MPC-Hybrid Model . . . 17

5 Auditable Quantum Authentication **17**
5.1 Protocol . . . 18
5.2 Security . . . 19
5.3 Generalized Online Input . . . 22
5.4 Compatibility of Quantum Authentication Code . . . 24

6 BoBW-MPQC-PVIA in the Preprocessing Model **25**
6.1 Input Encoding and Ancilla Preparation . . . 25
6.2 Distributed Circuit Evaluation . . . 28
6.3 Multiparty Quantum Computation . . . 32

7 BoBW-MPQC-PVIA without Trusted Setup **33**

8 Extentions **35**
8.1 Constant Round MPQC-PVIA in the Preprocessing Model . . . 35
8.2 MPQC-PVIA from Trap-code AQA . . . 36

1 Introduction

Validating an accusation is a challenge in our daily lives, from court to cryptographic applications. It could be extremely harmful if either an actual malevolent behavior or a false accusation occurs. Hence, giving sufficient evidence to a trusted party or the general public becomes an important issue. Consider the following scenario: Alice, a food delivery driver, picks up a Bento Box from a Japanese restaurant and delivers it to Bob, the recipient. If Alice and Bob are both truthful, they will be satisfied. But what if one of them is malicious? For example, Alice may open the lunchbox, eat a bit, and then send it to Bob. Or Bob simply refuses to pay and accuses Alice of not delivering it to him. It turns out to be "Rashomon". If they end up going to court, the judge or the jury in court lacks adequate evidence and eyewitness testimony to reach a judgment. This is a toy example of *public verifiability* i.e., everyone, including non-participants such as the jury, can verify who the malicious party is [BDO14, BOSSV20].

This deliveryman problem can happen to *quantum message transmission* (QMT), where someone has to deliver a quantum message. Thanks to the long arm of quantum law, we are able to present a generic transformation, *Auditable Quantum Authentication* (AQA), to deal with it. In our AQA, Alice must take responsibility for her delivery, and everyone will receive a checksum for verification. The judge and jury (outside observer with only classical computational power) can now verify the entire process independently in the event of a dispute. Although AQA can realize public verifiable identifiable abort QMT, more sophisticated tasks, including multiparty computation (MPC), are preferable.

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 in 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 are in the fully quantum setting i.e., the functionality, including inputs and outputs, are quantum. While [CGS02, BOCG+06] achieved full security with an honest majority, it is impossible to obtain MPQC with full security in general without an honest majority [ABDR04]. To circumvent this impossibility, MPQC protocols with a dishonest majority, such as [DNS12, DGJ+20, BCKM21], seek the weaker notion of security with abort, which allows all honest parties to abort when they observe an attack. However, such a notion is vulnerable to a *denial-of-service attack* because an attacker can repeatedly invoke aborts. In light of this, [ACC+21] proposed an MPQC protocol with identifiable abort (MPQC-SWIA), allowing all honest parties to agree on an identity of a corrupted party as an output in case of an abort. Still, their identification mechanism only allows inside MPQC participants to identify a malicious player. Returning to the deliveryman problem from earlier, MPQC-SWIA is unable to provide sufficient evidence to the jury, who is absent from this matter. Given the state of affairs, we ask:

Is it possible to construct MPQC with public verifiable identifiable abort?

In the classical setting, one can turn MPC-SWIA into MPC-PVIA. Obtaining a public verifiable protocol requires each participant to broadcast committed messages to outsider observers. However, it is impossible to broadcast quantum states due to the no-cloning theorem, so the classical technique cannot be applied in the quantum setting. A recent line of work that also aims for public verifiability is classical verification of quantum computation (CVQC) [Mah18]. However, due to its restriction to quantum computation that is performed by a single server with classical outputs, CVQC cannot fit in general MPQC.

It is widely believed that the no-cloning theorem and the traditional sender-receiver mechanism in the quantum authentication code prevent quantum techniques from realizing an MPQC-PVIA protocol. As a result, we design a transformation for public verifiability that subverts the traditional sender-receiver rationale. We will close the gap in our technical overview section later on.

Along the line of dishonest majority setting, cryptographers have considered MPQC against $n-1$ malicious parties among n participating parties with some sort of security with abort because full security is impossible in this case. Another interesting setting is to reduce the maximal number of malicious parties while conditionally requiring full security. Such a notion is called best-of-both-worlds (BoBW) security[1]. In the classical setting, [IKK+11] constructs for every $\mathtt{t} \leq \lfloor \frac{n-1}{2} \rfloor$ an MPC protocol that achieves security with abort against $n-1-\mathtt{t}$ malicious parties and achieves full security tolerating $\mathtt{t}$ malicious parties. [Kat07] proved that these corruption thresholds are optimal. In the quantum setting, none of the existing MPQC protocols can be extended to satisfy BoBW security. It is unclear in MPQC whether best-of-both-worlds security is achievable. 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

- **Generic Transformation:** We provide a generic AQA transformation that equips quantum authentication codes with public verifiability. The transformation works with various authentication codes, including Clifford and Trap codes [ABOEM17, BGS13]. It opens up the feasibility of constructing MPQC protocols using the Trap authentication code, 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].

- **Best-of-Both-Worlds MPQC-PVIA:** We build the first best-of-both-worlds MPQC-PVIA protocol obtaining either full security or security with public verifiable identifiable abort. Our construction is based on our AQA transformation and MPQC-SWIA. The key to arriving at best-of-both-worlds security is our protocol's compatibility with decentralized quantum computation. In particular, no single participant 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].

[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.

- **Classical Online Communication** Our MPQC protocol consists of an offline phase and an online phase. Our MPQC protocol requires just classical communication during the online phase, whereas many previous MPQC protocols [DGJ+20, ACC+21] require a quantum network.

Following [DNS12, DGJ+20], our protocol assumes a classical MPC-PVIA as an ideal functionality, and we refer the reader to Section 4.2 for more details on the model. Our protocol is proven under the standard real/ideal paradigm.

Theorem 1.1. *There exists a best-of-both-worlds multi-party quantum computation protocol secure with public verifiable identifiable abort of threshold* $\mathtt{t}$ *over poly-size quantum circuits for every* $\mathtt{t} \leq \lfloor \frac{n-1}{2} \rfloor$ *in the preprocessing MPC-hybrid model .*

We then remove the trusted preprocessing for circuits with only private outputs.

Theorem 1.2. *There exists a best-of-both-worlds multi-party quantum computation protocol secure with public verifiable identifiable abort of threshold* $\mathtt{t}$ *over poly-size quantum circuits with only private outputs for every* $\mathtt{t} \leq \lfloor \frac{n-1}{2} \rfloor$ *in the MPC-hybrid model.*

1.2 Open Problems

- **MPQC-PVIA with different security.** Our protocols have been proven in the real/ideal paradigm. MPQC-PVIA with universally composable (UC) security remains an unsolved topic. In addition, our protocols only consider static adversaries i.e., all the malicious players are decided before the game starts. MPQC-PVIA against adaptive adversaries remains unclear.

- **Trap-code form MPQC-PVIA.** Existing MPQC with dishonest majority protocols are mostly in Clifford Authentication Code. Using our Trap-form AQA to create Trap-code form MPQC protocols with desirable properties could also be a good open problem.

- **Applications of AQA.** There are many public verifiable primitives in classical cryptography, but their quantum counterparts are unknown. Our AQA may facilitate the development; for example, public verifiable quantum fully homomorphic encryption (pvQFHE).

2 Technical overview

In this section, we explain the reason public verifiability does not follow directly from existing works. Then, we introduce our new primitive, Auditable Quantum Authentication (AQA), which plays a crucial role in sending a quantum message securely. Afterward, we show how to apply AQA to construct MPQC-PVIA. At the end of the section, we discuss the difficulty and our method of achieving best-of-both-worlds (BoBWs) security.

2.1 Quantum Background

Before our discussion, we recall some quantum backgrounds for unfamiliar readers. Those who are experienced in quantum computing can skip this part.

Quantum States and Gates We use lowercase Greek alphabets, e.g., ρ, σ, τ, to denote the density matrix of a quantum state. The notation $U \cdot \rho$ stands for the result $U\rho U^\dagger$ of applying a unitary transformation U to a quantum state ρ. The bit flip gate $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, phase flip gate $Z = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ and bit-phase flip gate $Y = \left[\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right]$ generates the single-qubit Pauli group $\mathscr{P}_1$. The n-qubit Pauli group $\mathscr{P}_n$ is the n-fold tensor of $\mathscr{P}_1$. The n-qubit Clifford group $\mathscr{C}_n$ is the normalizer of $\mathscr{P}_n$ among the group of unitary transformations.

Quantum One-Time Pad A quantum one-time pad is a symmetric encryption scheme that provides information-theoretic security. It is very similar to the classical one-time pad. The key space is the Pauli group $\mathscr{P}_n$. Encryption of an n-qubit state ϕ under $P \in \mathscr{P}_n$ results in the ciphertext $P \cdot \phi$.

Quantum Teleportation Quantum teleportation transmits quantum information from sender to receiver by sending only classical messages. The sender and the receiver pre-share an EPR pair (e_S, e_R). The sender holds an input state ϕ and half of the EPR pair e_S, executes the teleportation circuit that ends with measurements, and sends the classical measurement outcomes z and x to the receiver for recovery. The receiver applies $Z^z X^x$ to the other half of the EPR pair e_R to obtain ψ.

Quantum Authentication Code Quantum authentication code prevents adversarial parties from making unauthorized alterations to the data. A specific code is the Clifford authentication code, whose encoding process augments a quantum state ϕ into a larger quantum state with traps, $\phi \otimes 0^t$, and applies a secret Clifford key E on it. No one can successfully alter the resulting state without knowing the Clifford key E. The key holder can tell whether the quantum state was tampered with by applying $E^\dagger$ and checking the trap measurement result.

Quantum Error Correction Code Quantum error correction code (QECC) protects quantum states from errors. QECC works by encoding a logical qubit in a particular entangled way into physical qubits, which are resilient against the effects of noise as a whole. The physical qubits can later be decoded to recover the original logical qubit. One usually prefers QECC that supports transversal computations, i.e., certain computations on logical qubits can be realized as parallel computations on the corresponding physical qubits.

Universal Quantum Computation A set of quantum gates realizes universal quantum computation if they can approximate every unitary operator. Sometimes, a gate can be reduced to other gates with the help of auxiliary states. In this paper, we will take the set of transversal gates of a QECC together with magic states to fulfill universal quantum computation. An example is to take the Clifford group with T magic states.

2.2 Public Verifiability is Hard to Achieve on MPQC

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. When adapting to MPQC in the fully quantum setting, there are several issues. If one considers using classical commitments to quantum messages [Mah18], one cannot fulfill MPQC that has purely quantum outputs because such classical commitment schemes always end with measurements. Instead, one may have to consider quantum commitments, which are rather lacking in the literature. Even if there are quantum commitments, they 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 that have a classical description. They cannot prove relations involving quantum commitments.

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

One may try to build MPQC-PVIA upon MPQC-SWIA protocols, 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 participants to agree on the same malicious party when protocol aborts, but an outside observer only sees two groups of people accusing each other. This is a result of how quantum authentication codes have typically been used: the sender encrypts a message with a random key before sending it to the receiver, who then checks to see if the checksum or traps match the consented randomness. This type of standard sender-receiver mechanism, however, has no advantage in terms of public verifiability because both "sending" and "receiving" are behaviors that must be validated, and both sender and receiver have the right to make accusations. One or both of them may be lying, but there is no rule to separate these cases and make the right judgment.

2.3 Our Solution: Auditable Quantum Authentication (AQA)

In previous MPQC works [DGJ+20, ACC+21, BCKM21], the receiver of a quantum authenticated message is in charge of detecting malicious actions of the sender. The issue is that the receiver is unable to offer public evidence for the sender's misdeeds of causing an authentication failure. To address this, we abandon the conventional sender-receiver pattern, and focus only on whether the sender sends an untampered message. To this end, we introduce Auditable Quantum Authentication (AQA), a generic transformation that works with many types of quantum authentication codes. The primary goal of AQA is to hold the sender of an authenticated message accountable for its actions via a test while keeping the receiver in place. There are two requirements of AQA. First, the receiver should always obtain a legitimate authenticated message (lunchbox) when the sender (deliveryman) passes the test. Second, everyone should learn that the sender is malicious if the test fails. With these guarantees, everyone can identify the exact party that attacks an AQA-based protocol.

We can integrate quantum teleportation and quantum message authentication scheme into an AQA scheme in the preprocessing model. A high-level idea is as follows. In the preprocessing phase, the trusted setup will generate EPR pairs with embedded secret keys, and distribute them to the sender and the receiver while giving the keys to classical MPC. The whole component is called portals.

Here, we divide the portals into three categories: sending portal, receiving portal, and checking portal. The sender can teleport the desired authenticated message using the sending portal. If the sender reports a valid teleportation measurement result, the receiving portal enables the receiver to retrieve a newly authenticated message. The checking portal, which the sender must measure and publish, indicates whether the sender has altered the teleportation measurement result or not. The actual detection process is carried out by classical MPC, which knows the secret key and produces a publicly verifiable detection outcome.

Warm-up To build intuition, we provide a protocol sketch of AQA as follows. Please take notice that although we present our AQA under the Clifford authentication code, the readers are welcome to instantiate AQA with the Trap authentication code (5.4) instead.

In the preprocessing phase, the trusted setup will generate EPR pairs (e_S, e_R), random Clifford authentication keys E, E', and random Pauli keys P_S, P_C, P_R. The trusted party then pretends that e_R is a ciphertext under Clifford code and parses $(E^\dagger \cdot e_R)$ into the plaintext part μ and the trap part τ, and distributes $(\mathcal{S}, \mathcal{C}) = (P_S \cdot e_S, P_C \cdot \tau)$ to the sender and $\mathcal{R} = P_R \cdot (E' \cdot (\mu \otimes 0^t))$ to the receiver. The states $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ are what we refer to as portals.

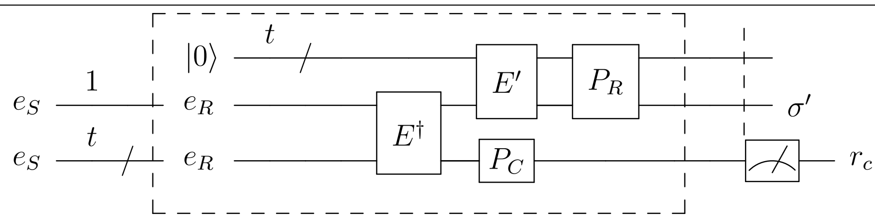

Figure 1: Auditable Quantum Authentication in Clifford code form.

In the online phase, the sender receives an additional quantum state $\sigma = E \cdot (\rho \otimes 0^t)$ for some ρ without knowing the key E. To remove E from σ, the sender can teleport σ through $\mathcal{S}$, and publishes the teleportation measurement result (z, x). If we look at the post-teleportation state of $E^\dagger \cdot e_R$, it is $E^\dagger X^x Z^z P_S^\dagger E \cdot (\rho \otimes 0^t)$. Since E is a Clifford gate, conjugation of a Pauli gate by E is again a Pauli, so the state can be rewritten as $(\hat{P}_\mu \cdot \rho) \otimes (\hat{P}_\tau \cdot 0^t)$ for some Pauli $\hat{P}_\mu, \hat{P}_\tau$. Using the fact that E' is also a Clifford gate, we can then derive the receiving portal as $\mathcal{R} = P_R \cdot (E' \cdot ((\hat{P}_\mu \cdot \rho) \otimes 0^t)) = \hat{P}_R E' \cdot (\rho \otimes 0^t)$ and the checking portal as $\mathcal{C} = P_C \hat{P}_\tau \cdot 0^t = \hat{P}_C \cdot 0^t$ for some Pauli $\hat{P}_R, \hat{P}_C$.

As soon as the receiver learns $\hat{P}_R$, it can obtain the state $\sigma' = E' \cdot (\rho \otimes 0^t)$ out of $\mathcal{R}$. The classical MPC can compute $\hat{P}_R$ for the receiver from $E', E, P_R, P_S, (z, x)$. However, the sender can deviate from the honest behavior either by altering σ or modifying (z, x), which corrupts the receiver's final state. To prevent this attack, we require that the sender also measures $\mathcal{C}$ and outputs the result r_c to the classical MPC. The expected measurement outcome of $\mathcal{C}$ depends on the traps within σ and the actual teleportation measurement result (z, x) and the secret random gate E, so verifying r_c acts as a randomized check on the sender's choice of σ and (z, x). In our analysis, we show that the check is sound so that the classical MPC can detect malicious actions of the sender effectively.

Although the above AQA requires a trusted setup in the preprocessing phase, we will later instantiate the preprocessing phase with MPQC-SWIA e.g., [ACC+21]. The resulting scheme becomes a two-stage protocol where PVIA is only guaranteed in the second stage. As we will see, this weaker security suffices for MPQC-PVIA as long as the circuit produces only private outputs.

2.4 From AQA to MPQC-PVIA

So far, we have developed AQA that achieves public verifiable identification in the context of quantum message transmission. Here, we describe how to use AQA to fulfill the task of MPQC while achieving PVIA. We proceed in two steps, by first assuming a trusted setup, and then removing this extra assumption.

Preprocessing MPQC-PVIA At the start of MPQC, the parties have to authenticate their inputs using keys known only by classical MPC. To do this in a PVIA manner, we resort to an input-independent trusted setup. The trusted setup can provide everyone with EPR pairs, half of which are authenticated using some secret keys. During the online phase, each party teleports its input to convert it into an authenticated message. We call this step input encoding (IE). So now, we have IE to encode our inputs and AQA to transmit message ciphertexts.

Another vital component of MPQC is performing computation on authenticated data, which is supported by many quantum authentication codes on a set of gates and measurements. For instance, using the Clifford authentication code, Clifford gates can be performed on a single ciphertext without any interaction. Furthermore, one can obtain verifiable measurement results on Clifford ciphertexts, as shown by [DGJ+20]. Thus, the parties can perform Clifford gates and measurements on their authenticated joint inputs. More generally, the protocol can allow universal quantum computation if the trusted setup additionally produces and distributes authenticated magic states in advance.

We now move on to examine security. In our protocol, each quantum computation unit is performed by a single party, so attacks during the computation step would only ruin the authentication code held by the attacker. The attacker will ultimately be identified when asked to transmit the message in a subsequent AQA. As a result, a protocol combining IE, AQA, and computation can achieve MPQC while maintaining PVIA security.

The protocol sketch of our preprocessing MPQC-PVIA is as follows. First, all parties execute IE so that a delegated party [2], say P_1, obtains the authenticated joint input. Next, P_1 performs computation instructed by classical MPC and executes AQA to send the authenticated outputs back. Finally, the classical MPC reveals the authentication keys to each party, and each party decodes its output. Whenever AQA fails, the classical MPC publicly outputs the sender's identity and halts the protocol.

MPQC-PVIA Next, we show how to obtain an MPQC-PVIA protocol by combining our preprocessing MPQC-PVIA with the MPQC-SWIA protocol of [ACC+21] and removing the trusted setup assumption. However, we must restrict ourselves to computing circuits with only private outputs. In this case, public verifiability only applies to verifying the malicious identity.

Recall that when MPQC-SWIA aborts, the classical MPC will output a partition of parties with all honest parties staying in the same division. Our approach is to run MPQC-SWIA hierarchically to prepare the preprocessing data needed for the preprocessing MPQC-PVIA. 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 run their instance of the preprocessing MPQC-PVIA protocol. With the guarantees of MPQC-SWIA and preprocessing MPQC-PVIA, the whole protocol achieves MPQC-PVIA.

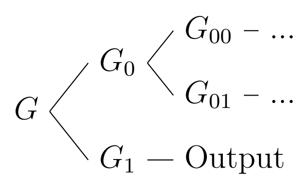

Figure 2: Hierarchical MPQC-SWIA Players try to generate portals and ancilla states with MPQC-SWIA. G contains the total players in the MPQC game. When the first MPQC-SWIA terminates with a failure, players separate into two groups G_0 and G_1. Then, players in G_0 and players in G_1 run MPQC-SWIA with players in their group. In this figure, G_1 completes the MPQC-SWIA task successfully and obtains output. Players in group G_1 can proceed to execute preprocessing MPQC-PVIA.

2.5 Difficulty of 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 also achieves full security against at most $\mathsf{t} \leq \lfloor \frac{n-1}{2} \rfloor$ corruptions as long as we require security with public verifiable identifiable abort against at most $n-1-\mathsf{t}$ malicious parties. In contrast, none of the existing MPQC protocols satisfy BoBW security.

The two worlds of honest majority and dishonest majority were once separated because of a tension between spreading information and extracting information. We elaborate on it as

[2]There could be multiple delegated parties, but here we assign only one party for clear presentation.

follows. MPQC protocols that obtain full security in an honest majority setting [BOCG⁺06] are based on quantum secret sharing (QSS) [CGL99]. The problem is that the secret shares sent between malicious parties are private information. Once the number of corrupted players reaches one-half, the simulator has no way to extract the adversary's input from the available secret shares. It also explains why existing MPQC protocols in the dishonest majority setting require each quantum message to pass through all parties. In this case, the quantum message must have gone through some honest player, and the simulator can extract input at this moment. As a result, these protocols are unable to divide every piece of 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 combine the hierarchical MPQC-SWIA with quantum error correction codes (QECC), which is a relaxation of QSS. In the beginning, parties use the hierarchical MPQC-SWIA to prepare QECC codewords that encode EPR pairs and distribute every codeword evenly to the parties. Afterward, the parties can perform distributed computation over QECC codewords. Full security can be achieved when only a few qubits of QECC are in the control of corrupted parties. One can view the hierarchical MPQC-SWIA as a vital piece of machinery that allows information extraction while contributing to the setup for distributed computation.

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

3 Preliminary

Let $[n] = \{1, \cdots, n\}$. For a vector v, its first n elements form the vector $v[:n]$, and the remaining form $v[n:]$. 1-norm is denoted as $||v||_1$. Given a set S, $s \leftarrow S$ means that s is uniformly sampled from S. A function $f\colon \mathbb{N} \to [0,1]$ is called negligible, if for every polynomial $p(\cdot)$ and all sufficiently large n, it holds that $f(n) < \frac{1}{|p(n)|}$. We use $\mathrm{negl}(\cdot)$ to denote an unspecified negligible function. QPT stands for quantum polynomial-time.

Quantum Notations Density matrices are written in lowercase Greek alphabets, e.g., ρ, σ, τ. The set of density matrices over the Hilbert space of n qubits is $\mathcal{D}^n$. The notation (ρ, σ) denotes a state on two registers that may be entangled. We sometimes write $0^n \in \mathcal{D}^n$ to represent $|0\rangle\langle 0|^{\otimes n}$. The shorthand $U \cdot \rho$ stands for the result $U\rho U^\dagger$ of applying a unitary U to ρ. Measurement in the computational basis is denoted by Π. States $g(n), h(n) \in \mathcal{D}^{\mathrm{poly}(n)}$ are statistically indistinguishable, denoted $g \approx h$, if they have trace distance $\mathrm{tr}\,|g(n) - h(n)| = \mathrm{negl}(n)$. States $g(n), h(n)$ are computationally indistinguishable, denoted $g \approx_c h$, if for any QPT distinguisher D outputting 1 bit, it holds that $||D(g(n)) - D(h(n))||_1 = \mathrm{negl}(n)$.

The Pauli group $\mathscr{P}_n$ acting on n qubits is the n-fold tensor of the group generated by $X = \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]$, $Z = \left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right]$ and $i\mathbb{1} = \left[\begin{smallmatrix}i&0\\0&i\end{smallmatrix}\right]$. An element $P \in \mathscr{P}_n$ can be written as $P = \pm i \bigotimes_{j\in[n]} X^{x_j} Z^{z_j}$. We denote the X, Z exponents of a Pauli P as $x(P), z(P)$. The Clifford group $\mathscr{C}_n$ acting on n qubits is the normalizer of $\mathscr{P}_n$. That is, $C \in \mathscr{C}_n$ 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 operation, we can swap the order where a Clifford and a Pauli are applied. An EPR pair (e_S, e_R) of length n is the first and second halves of $|\Phi^+\rangle\langle\Phi^+|^{\otimes n}$ where $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The abbreviation $\mathsf{CNOT}_{(b_1,\cdots,b_n)}{}^{R_0,\cdots,R_n}$ stands for the composition $\mathsf{CNOT}^{b_1}{}^{R_0,R_1}\cdots\mathsf{CNOT}^{b_n}{}^{R_0,R_n}$.

3.1 Quantum Teleportation

Definition 3.1 (Quantum Teleportation)**.** 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:

- $\mathsf{TP.Send}(\psi^M, e_S{}^S) := \Pi^{M,S} H^M \mathsf{CNOT}^{M,S}(\psi, e_S)$ which yields strings (z, x).
- $\mathsf{TP.Receive}(z, x, e_R) :=$ Apply $(X^x Z^z)^\dagger$ on e_R which yields the state ψ.

Define also the unitary $\mathsf{TP}^{M,S} := H^M \mathsf{CNOT}^{M,S}$ for sending without measurement. It always holds that $\mathsf{TP.Receive}(\mathsf{TP.Send}(\psi, e_S), e_R) = \psi$ and $\mathsf{TP.Send}$ has uniformly random outputs.

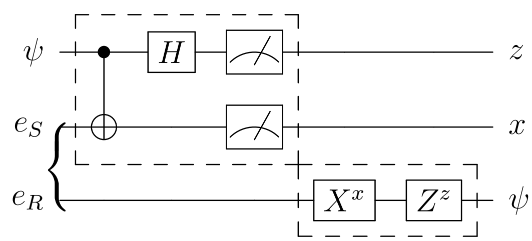

Figure 3: single-qubit quantum teleportation

3.2 Quantum One-Time Pad

Definition 3.2 (Quantum One-Time Pad (QOPT))**.** Let $\rho \in \mathcal{D}^n$. A QOPT with keys $x, z \in \{0,1\}^n$ is a symmetric-key encryption scheme that consists following two algorithms.

- Encryption
$$\mathsf{QOPT.Enc}(\rho) := (X^x Z^z)\rho(X^x Z^z)^\dagger$$
- Decryption
$$\mathsf{QOPT.Dec}(\rho) := (X^x Z^z)^\dagger\rho(X^x Z^z)$$

Lemma 3.3 (Security of QOPT)**.** *The QOTP is information-theoretic secure ,i.e., the ciphertext is maximally mixed to a player without keys.*

$$\frac{1}{2^n}\sum_{x,z}(X^x Z^z)\rho(X^x Z^z)^\dagger = (\frac{\mathbb{1}}{2})^{\otimes n}$$

3.3 Twirling

Twirling [BDSW96] of a state ρ is to perform a random operator U among a group G of unitary operators on ρ, and tracing out U. It is defined as the CPTP map

$$\mathcal{T}_G(\rho) := \frac{1}{|G|}\sum_{U\in G} U \cdot \rho.$$

Twirling a state over Pauli group or Clifford group results in the fully mixed state $(\frac{\mathbb{1}}{2})^{\otimes n}$. That is, $\mathcal{T}_{\mathscr{P}}(\rho) = \mathcal{T}_{\mathscr{C}}(\rho) = (\frac{\mathbb{1}}{2})^{\otimes n}$. On the other hand, twirling of an operator V is defined as the CPTP map

$$\mathcal{T}_G(V) := \frac{1}{|G|}\sum_{U\in G} U^\dagger V U \cdot \rho.$$

The following lemma on Pauli twirl implies that any quantum attack against quantum one-time pad is equivalent to a probabilistic combination of Pauli attacks.

Lemma 3.4 (Pauli Twirl)**.** *Let $\rho \in \mathcal{D}^n$. For every Pauli operators P, P', it holds that*

$$\frac{1}{4^n}\sum_{x,z\in\{0,1\}^n} (X^x Z^z)^\dagger P (X^x Z^z) \rho (X^x Z^z)^\dagger P'^\dagger (X^x Z^z) = \begin{cases} P\rho P^\dagger, & \textit{if } P = P' \\ 0, & \textit{otherwise.}\end{cases}$$

3.4 Quantum Error-Correcting Codes

Quantum error correcting code protects quantum states from errors. The encoding algorithm $\mathsf{QECC.Enc}$ of a $[[q, 1, d]]$ QECC code encodes a 1-qubit message into a q-qubit codeword. Through the decoding procedure $\mathsf{QECC.Dec}$, the QECC code can correct arbitrary errors on $\lfloor\frac{d-1}{2}\rfloor$ qubits, or $d-1$ erasure-errors on known positions.

Definition 3.5 (Quantum Error Correction Code)**.** A $[[q, 1, d]]$-QECC consists of an QPT encoding algorithm $\mathsf{QECC.Enc} : \mathcal{D}^1 \to \mathcal{D}^q$ and a QPT decoding algorithm $\mathsf{QECC.Dec} : \mathcal{D}^q \to \mathcal{D}^1$ such that for all quantum channel Δ acting on $\lfloor\frac{d-1}{2}\rfloor$ qubits and for all $\rho \in \mathcal{D}^1$, it holds that

$$(\mathsf{QECC.Dec} \circ \Delta \circ \mathsf{QECC.Enc})(\rho) = \rho.$$

We consider CSS codes in particular due to the following properties. First, the encoding process consists only of Clifford gates, which we denote by QECC. Second, they allow certain Clifford gates to be transversally computed over codewords. Third, measurement can be transversally performed over codewords by running the classical decoding algorithm of the linear code associated with the CSS code over the measurement outcome of codewords.

To arrive at best-of-both-worlds security for any threshold $\mathtt{t} \le \lfloor\frac{n-1}{2}\rfloor$, we can use quantum polynomial codes such as the quantum secret sharing scheme of [CGL99]. Such codes exist for every $q = 2d - 1$ over qudits, which can satisfy the condition $\frac{d-1}{q} > \frac{\mathtt{t}}{n}$ needed for full security against $\mathtt{t}$ corruptions. For simplicity, we present our main arguments over qubits, and set $q = n$ or any multiple of n.

3.5 Quantum Authentication Code

Quantum authentication code prevents adversarial parties from making unauthorized alterations to the data. Here, we introduce Clifford code [ABOEM17] and Trap code [BGS13]. Below, $\perp$ will denote a reject symbol. It is actually stored in a separate flag register, but for simplicity we will just put it in the original state register. Encoding of $\perp$ is always defined as $\perp$.

3.5.1 Clifford Code

Definition 3.6 (Clifford Code [ABOEM17])**.** Let M be the message register, T be the trap register. The set of keys is the Clifford group $\mathscr{C}_{|MT|}$. Define the projectors $P_{\mathsf{Acc}} := |0^t\rangle\langle 0^t|$ and $P_{\mathsf{Rej}} := \mathbb{1}^t - P_{\mathsf{Acc}}$. The Clifford authentication code $\mathsf{CliffordAuth} = (\mathsf{C.Enc}, \mathsf{C.Dec})$ is defined as follows.

- **Encoding**
$$\mathsf{C.Enc}_E(\rho^M) := E(\rho^M \otimes |0^t\rangle\langle 0^t|^T)E^\dagger.$$

- **Decoding**
$$\mathsf{C.Dec}_E(\sigma^{MT}) := \mathrm{tr}_T\left(P^T_{\mathsf{Acc}} E^\dagger \sigma^{MT} E\right) + \mathrm{tr}_{MT}\left(P^T_{\mathsf{Rej}} E^\dagger \sigma^{MT} E\right) \perp^M.$$

In other words, Clifford code augments the message state ϕ^M with traps $|0^t\rangle\langle 0^t|^T$, and applies a secret random Clifford gate E on them. The decoding procedure is to apply $E^\dagger$ followed by measuring the register T. If the measurement result are not all zero, the content in M is replaced with $\perp$. The scheme is secure in the sense that the authentication result is either the original state or a rejection.

Theorem 3.7 (Security of Clifford Code, Appendix C of [DGJ+20])**.** *For all CPTP map $\mathcal{A}^{MTR}$ there exists two CP maps $\mathscr{A}^R_{\mathsf{Acc}}$ and $\mathscr{A}^R_{\mathsf{Rej}}$ satisfying $\mathscr{A}^R_{\mathsf{Acc}} + \mathscr{A}^R_{\mathsf{Rej}}$ is trace preserving, such that for any input ρ^{MR} it holds that*

$$\left\| \mathop{\mathbb{E}}_{k\in\mathscr{C}_{|MT|}} [\mathsf{Dec}_k(\mathcal{A}^{MTR}\mathsf{Enc}_k(\rho))] - \left(\mathscr{A}^R_{\mathsf{Acc}}(\rho) + |\perp\rangle\langle\perp|^M \otimes \mathrm{tr}_M[\mathscr{A}^R_{\mathsf{Rej}}(\rho)]\right) \right\| \leq \mathrm{negl}\,|T|$$

3.5.2 Trap Authentication Code

Definition 3.8 (Trap Code [BGS13])**.** The key set of a trapcode that uses $[[t,1,d]]$-QECC is $Sym_{3t} \times \{0,1\}^{3t} \times \{0,1\}^{3t}$. Let M be the 1-qubit register, $\tilde{M}$ be the QECC codeword register, $C = \tilde{M}T_XT_Z$ be the ciphertext register, where T_X and T_Z are trap registers. Define the projectors $P_{\mathsf{Acc}} := |0\rangle\langle 0|^{\otimes t} \otimes |+\rangle\langle +|^{\otimes t}$ and $P_{\mathsf{Rej}} := \mathbb{1}^{2t} - P_{\mathsf{Acc}}$. Trap authentication code $\mathsf{TAuth} = (\mathsf{T.Enc}, \mathsf{T.Dec})$ is defined as follows.

- **Encryption**
$$\mathsf{T.Enc}_{\Pi,x,z}(\rho^M) := X^xZ^z\Pi(\mathsf{QECC.Enc}(\rho^M) \otimes |0\rangle\langle 0|^{t^{T_X}} \otimes |+\rangle\langle +|^{t^{T_Z}})(X^xZ^z\Pi)^\dagger.$$

- **Decryption**

$$\mathsf{T.Dec}_{\Pi,x,z}(\sigma^{\tilde{M}T_XT_Z}) := \mathsf{QECC.Dec}(\mathrm{tr}_{T_XT_Z}\left(I^{\otimes t}P^T_{\mathsf{Acc}} \otimes (X^xZ^z\Pi)^\dagger \sigma^{\tilde{M}T_XT_Z}(X^xZ^z\Pi)\right))$$
$$\otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}| + |\bot\rangle\langle\bot|^M \mathrm{tr}_{\tilde{M}T_XT_Z}\left(P^T_{\mathsf{Rej}} \otimes (X^xZ^z\Pi)^\dagger \sigma^{\tilde{M}T_XT_Z}(X^xZ^z\Pi)\right).$$

In other words, Trap code augments the message state ϕ into a larger quantum state, $\mathsf{QECC.Enc}(\phi) \otimes |0\rangle\langle 0|^t \otimes |+\rangle\langle +|^t$, and applies a random permutation Π followed by a quantum one-time pad X^xZ^z. By checking the trap measurement result, trap code provides authentication against message tampering. The next theorem quantifies its security.

Theorem 3.9 (Security of Trap Code [BGS13])**.** *A trapcode using $[[t, 1, d]]$-QECC is a $\frac{2}{3}^{\frac{d}{2}}$-secure quantum authentication scheme i.e., it holds that for each fixed choice of $3n$-qubit Pauli operation P, the probability that P operates nontrivially on logical data but has no error syndrome is at most $(\frac{2}{3})^{\frac{d}{2}}$.*

4 Model and Definition

We focus on interactive protocols between n parties with quantum capabilities. They can communicate using pairwise quantum channels or a public 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 who passively receives and records classical information from the broadcast channel all the time. The adversary can statically corrupt up to $n-1$ parties. Sometimes we consider the preprocessing model, which allows an input-independent trusted setup to be executed prior to the actual protocol.

4.1 Best-of-Both-Worlds Multi-Party Quantum Computation Secure with Public Verifiable Identifiable Abort

A multi-party quantum computation protocol is defined using the real vs. ideal paradigm. In the ideal world, the parties delegate the task of interest to a trusted party T. Let $\mathsf{P}_1, \cdots, \mathsf{P}_n$ be the parties involved in the computation, and I be the set of corrupted parties controlled by an adversary $\mathcal{A}_I$. There is an observer O that receives public information from T. Let C be a quantum circuit taking $(\ell_1, \cdots, \ell_n)$ input qubits and ℓ_{anc} ancilla qubits, a total of $\ell = \Sigma_i \ell_i + \ell_{\mathsf{anc}}$ qubits, and outputs ℓ' qubits along with some classical string r'_{out}. The ideal functionality we would like to achieve for multi-party quantum computation over a circuit C is as follows.

$\mathsf{Ideal}^{\mathsf{MPQC}}$: Multi-party Quantum Computation Secure with Public Verifiable Identifiable Abort

Common Input:

The security parameter 1^κ.

Input:

P_i holds private input $\rho_i \in \mathcal{D}^{\ell_i}$.

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 samples ρ_{anc} and computes $(\rho'_1, \cdots, \rho'_n, r'_{\text{out}}) = C(\tilde{\rho}_1, \cdots, \tilde{\rho}_n; \rho_{\mathsf{anc}})$.

T sends back outputs:

T sends ρ'_i to all $\mathsf{P}_i \in I$.
P_i in I can send `abort` message to T. If so, T publicly aborts to P_i.
If there is no `abort`, T sends ρ'_i to all $\mathsf{P}_i \notin I$ and publicly outputs r'_{out}.

Outputs:

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 its view.

We also consider $\mathsf{Ideal}^{\mathsf{MPQC-Full}}$, the ideal world of MPQC with full security. It is obtained by not allowing any abort messages to take place in the ideal world $\mathsf{Ideal}^{\mathsf{MPQC}}$. We denote by $\mathsf{Ideal}^{\mathsf{MPQC}}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\kappa, C, \rho_1, \cdots, \rho_n)$ the joint output of the honest parties, the observer, and the adversary at the end of $\mathsf{Ideal}^{\mathsf{MPQC}}$ when it is run by T and $\mathsf{P}_i(\rho_i)$ in the presence of an adversary $\mathcal{A}_I(\rho_{\mathsf{aux}})$ corrupting parties in I. We also define $\mathsf{Ideal}^{\mathsf{MPQC-Full}}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\kappa, C, \rho_1, \cdots, \rho_n)$ in a similar way.

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

Definition 4.1. We say that a protocol Π is a multi-party quantum computation secure with public verifiable identifiable abort (MPQC-PVIA) over a circuit C, if for every non-uniform (QPT) adversary $\mathcal{A}_I$ corrupting I, there is a non-uniform (QPT) simulator Sim_I corrupting I, such that for any quantum inputs $\rho_i \in \mathcal{D}^{1_i}$,

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

Furthermore, we say that a protocol Π is a best-of-both-worlds multi-party quantum computation secure with public verifiable identifiable abort (BoBW-MPQC-PVIA) of threshold t if the indistinguishability condition is modified as

$$\mathsf{Real}^{\Pi}_{\mathcal{A}_I(\rho_{\mathsf{aux}})} \approx \mathsf{Ideal}^{\mathsf{MPQC-Full}}_{\mathsf{Sim}_I(\rho_{\mathsf{aux}})}(1^\kappa, C, \rho_1, \cdots, \rho_n) \text{ when } |I| \leq \mathsf{t}$$
$$\mathsf{Real}^{\Pi}_{\mathcal{A}_I(\rho_{\mathsf{aux}})} \approx \mathsf{Ideal}^{\mathsf{MPQC}}_{\mathsf{Sim}_I(\rho_{\mathsf{aux}})}(1^\kappa, C, \rho_1, \cdots, \rho_n) \text{ when } |I| \leq n - 1 - \mathsf{t}$$

We can also define these versions of multiparty quantum computation for protocols (Σ, Π) with trusted setup. One simply replaces $\mathsf{Real}^{\Pi}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}$ with $\mathsf{Real}^{\Pi\circ\Sigma}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}$ in the definitions.

4.2 MPC-Hybrid Model

Following [DNS12, DGJ+20, ACC+21, BCKM21], we assume a *reactive*[3] BoBW-MPC-PVIA ideal functionality cMPC for classical multiparty computation within our MPQC protocol. We refer this setting as the MPC-hybrid model. The ideal world of MPC is similar to that of MPQC defined in section Section 4.1, but allows only classical messages and classical computations. We will simply view cMPC as a trusted classical party. One can also instantiate the BoBW-MPC-PVIA ideal functionality using a post-quantum universally-composable BoBW-MPC-PVIA protocol. A concrete construction of such a protocol is to plug the universally-composable MPC-PVIA 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

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 its sending action. AQA is designed to identify the malicious sender only, while the receiving behavior is automatically guaranteed by successfully passing the test. For simplicity, we present Clifford-code form AQA. One can feel free to replace it with Trap-code form AQA (Figure 4). We now formally define the ideal world of AQA. Let T be the trusted party, $\mathsf{P_s}$ be the sender, $\mathsf{P_r}$ be the receiver, and O be the observer.

$\mathsf{Ideal}^{\mathsf{AQA}}$: Ideal World of Auditable Quantum Authentication

Common Input:
The security parameter 1^κ and the number of traps $t = t(\kappa)$.

Input:
T holds key $E \in \mathscr{C}_{1+t}$, and $\mathsf{P_s}$ holds ciphertext $\sigma = E \cdot (\rho \otimes 0^t) \in \mathcal{D}^{1+t}$.

T receives quantum state
$\mathsf{P_s}$ sends σ to T. If $\mathsf{P_s}$ is corrupted, it can send $\perp$ to T instead.

T re-authenticates and forwards the quantum state
If T receives $\perp$ from $\mathsf{P_s}$, it publicly aborts to $\mathsf{P_s}$.
Otherwise, T sends $\sigma' = E'E^\dagger \cdot \sigma$ to $\mathsf{P_r}$ using fresh key $E' \leftarrow \mathscr{C}_{1+t}$.

Output:
$\mathsf{P_s}, \mathsf{P_r}, \mathsf{O}$ output whatever they received from T.
T outputs E', where E' is replaced with $\perp$ upon abort.
$\mathcal{A}$ outputs its view.

We denote by $\mathsf{Ideal}^{\mathsf{AQA}}_{\mathcal{A}(\rho_{\mathsf{aux}})}(1^\kappa, E, \rho)$ the joint output of $\mathsf{Ideal}^{\mathsf{AQA}}$ run by $\mathsf{T}(E)$, $\mathsf{P_s}(E \cdot (\rho \otimes 0^t))$, $\mathsf{P_r}$ in the presence of an adversary $\mathcal{A}(\rho_{\mathsf{aux}})$ that may corrupt both $\mathsf{P_s}$ and $\mathsf{P_r}$.

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

5.1 Protocol

To realize the ideal functionality of AQA, we first consider a preprocessing protocol where the sender indeed gets a ciphertext $\sigma = E \cdot (\rho \otimes 0^t)$ as online input. In section 5.3, we show how the protocol can be generalized to deal with a somewhat twisted online input.

The whole procedure of our AQA protocol consists of two parts: a trusted offline phase and an online protocol. In the offline phase, the trusted setup Σ^{AQA} generates triple portals by applying quantum one-time pad encryption and re-authentication operation on EPR pairs. The trusted setup stores the keys in cMPC. The sender $\mathsf{P_s}$ receives the sending portal $\mathcal{S}$ and the checking portal $\mathcal{C}$, whereas the receiver $\mathsf{P_r}$ receives the receiving portal $\mathcal{R}$.

Based on this pre-shared information, the online protocol Π^{AQA} goes as follows. The sender $\mathsf{P_s}$ teleports his/her ciphertext input σ via the sending portal $\mathcal{S}$ and measures the checking portal $\mathcal{C}$. There are two steps for cMPC to check the consistency of the sender's teleportation result (r_z, r_x) and measurement result r_c. First, cMPC XORs (r_z, r_x, r_c) with (s_z, s_x, s_c) to cancel out the effect of quantum one-time pads present in $\sigma, \mathcal{S}, \mathcal{C}$. Second, cMPC verifies whether the resulting values satisfy a conjugation relation defined by the secret key E. Any inconsistency indicates that the sender is malicious. On the other hand, cMPC can produce a correct key from consistent measurement results, which guarantees the authenticity of the receiver's receiving portal $\mathcal{R}$.

Protocol 1 $(\Sigma^{\mathsf{AQA}}, \Pi^{\mathsf{AQA}})$ for Auditable Quantum Authentication

---

Parameters: t

Preprocessing $\Sigma^{\mathsf{AQA}}(\mathsf{P_s}, \mathsf{P_r}, E)$**:**

1. Generate coordinate-wise EPR pair (e_S, e_R) of length $1 + t$.
2. Sample random Pauli gates $P_M, P_S, P_R \leftarrow \mathscr{P}_{1+t}$, $P_C \leftarrow \mathscr{P}_t$.
   Sample random authentication key $E' \leftarrow \mathscr{C}_{1+t}$.
3. Set $\tilde{E} = P_M^{\dagger} E$ and parse $\tilde{E}^{\dagger} \cdot e_R$ into (μ, τ) where $\mu \in \mathcal{D}^1$ and $\tau \in \mathcal{D}^t$.
4. Send to $\mathsf{P_s}$ the sending portal $\mathcal{S} = P_S \cdot e_S$ and the checking portal $\mathcal{C} = P_C \cdot \tau$.
   Send to $\mathsf{P_r}$ the receiving portal $\mathcal{R} = P_R E' \cdot (\mu \otimes 0^t)$.
   Send to cMPC the secret $(P_M, P_S, P_R, P_C, E, E')$.

Online Input:

1. $\mathsf{P_s}$ holds private online input $\sigma = E \cdot (\rho \otimes 0^t)$ for some $\rho \in \mathcal{D}^1$.

Protocol Π^{AQA}**:**

1. $\mathsf{P_s}$ teleports σ via sending portal $\mathcal{S}$ and measures traps on the checking portal $\mathcal{C}$, and then $\mathsf{P_s}$ sends $(r_z, r_x) \leftarrow \mathsf{TP.Send}(\sigma, \mathcal{S})$ and $r_c \leftarrow \Pi(\mathcal{C})$ to cMPC.
2. cMPC derives classical one-time pads $(s_z, s_x, s_c) = x(\mathsf{TP}(P_M \otimes P_S)\mathsf{TP}^{\dagger} \otimes P_C)$.
   cMPC decrypts the measurement result $(r'_z, r'_x, r'_c) \leftarrow (r_z, r_x, r_c) \oplus (s_z, s_x, s_c)$.
   cMPC parses $\tilde{E}^{\dagger} X^{r'_x} Z^{r'_z} \tilde{E} \in \mathscr{P}_{1+t}$ into $P_\mu \otimes P_\tau$ where $P_\mu \in \mathscr{P}_1$ and $P_\tau \in \mathscr{P}_t$.
   If $r'_c \neq x(P_\tau)$, cMPC publicly outputs $\mathsf{P_s}$ and all parties abort to $\mathsf{P_s}$.
   cMPC sends the correction Pauli $P' = P_R E'(P_\mu \otimes \mathbb{1}_t)E'^{\dagger}$ to $\mathsf{P_r}$.

Output:

1. $\mathsf{P_r}$ outputs $\sigma' = P'^{\dagger} \cdot \mathcal{R}$.
2. cMPC outputs E'.

5.2 Security

Lemma 5.1. *For every non-uniform* (QPT) *adversary $\mathcal{A}$ corrupting player set I, there is a non-uniform* (QPT) *adversary $\mathsf{Sim}_{\mathcal{A}}$ corrupting I, such that for any states $\rho, \rho_{\mathsf{aux}}$,*

$$\{\mathsf{Real}^{\Pi^{\mathsf{AQA}} \circ \Sigma^{\mathsf{AQA}}}_{\mathcal{A}(\rho_{\mathsf{aux}})}(1^{\kappa}, E, \rho) | E \leftarrow \mathscr{C}_{1+t}\} \approx \{\mathsf{Ideal}^{\mathsf{AQA}}_{\mathsf{Sim}_{\mathcal{A}(\rho_{\mathsf{aux}})}}(1^{\kappa}, E, \rho) | E \leftarrow \mathscr{C}_{1+t}\}$$

We split the security argument of our AQA protocol into two parts. In the first part, we analyze for any adversary the final state of the real protocol. In the second part, we show for any adversary a simulator that produces an indistinguishable final state. Our simulation technique is similar to that of [BW16].

Analysis of the Real World The following analysis will hold for every E'. Denote by (M, S, C, R, W) the registers that are used to store $(\sigma, \mathcal{S}, \mathcal{C}, \mathcal{R}, \rho_{\mathsf{aux}})$. In the most general case, the adversary has access to all these registers. Without loss of generality, it suffices to consider attacks that happen in protocol step 1 of the form $\Pi^{M,S,C} A^{M,S,C,R,W} \mathsf{TP}^{M,S}$ for some completely positive map $A : \rho \mapsto A \rho A^{\dagger}$. After the attack is applied, cMPC learns the values (r_z, r_x, r_c) stored in (M, S, C) and decides whether to abort the sender. We analyze the final state when cMPC does not abort the sender as follows. First, we simplify the state before attack A is applied.

$$\begin{aligned}
&\mathsf{TP}^{M,S} \cdot (\sigma, \mathcal{S}, \mathcal{C}, \mathcal{R}) \\
=&\mathsf{TP}^{M,S} {P_M}^{M} {P_S}^{S} {P_C}^{C} P_R {E'}^{R} \pi^{C,R} \cdot (\tilde{E} \cdot (\rho \otimes 0^t)^{M}, {e_S}^{S}, ((\tilde{E}^{\dagger} \cdot e_R) \otimes 0^t)^{C,R}) \\
=&P^{M,S,C} P_R {E'}^{R} \pi^{C,R} \mathsf{TP}^{M,S} \cdot (\tilde{E} \cdot (\rho \otimes 0^t)^{M}, {e_S}^{S}, ((\tilde{E}^{\dagger} \cdot e_R) \otimes 0^t)^{C,R})
\end{aligned} \tag{1}$$

where $\pi^{C,R}$ is the operation that exchanges the first qubit with the following t qubits, i.e., $(\mu, \tau, \xi) \mapsto (\tau, \mu, \xi)$ for $\mu \in \mathcal{D}^1$ and $\tau \in \mathcal{D}^t$. The Pauli $P^{M,S,C}$ in (1) is the conjugation of ${P_M}^{M} {P_S}^{S} {P_C}^{C}$ by the Clifford operation $\mathsf{TP}^{M,S}$. Then we expand the teleportation operator in (1) as

$$\begin{aligned}
&\mathsf{TP}^{M,S} \cdot (\tilde{E} \cdot (\rho \otimes 0^t)^{M}, {e_S}^{S}, ((\tilde{E}^{\dagger} \cdot e_R) \otimes 0^t)^{C,R}) \\
=&\frac{1}{2^N} \sum_{\alpha,\beta,\alpha',\beta'} |\beta\rangle \langle\beta'|^{M} \otimes |\alpha\rangle \langle\alpha'|^{S} \otimes ((\tilde{E}^{\dagger} X^{\alpha} Z^{\beta} \tilde{E} (\rho \otimes 0^t) \tilde{E}^{\dagger} Z^{\beta'} X^{\alpha'} \tilde{E}) \otimes 0^t)^{C,R}
\end{aligned}$$

where $N = 2(1+t)$. To ease notation, we will write $\delta = (\beta, \alpha)$ and define the Pauli $P_{\tilde{E},\delta} = (\tilde{E}^{\dagger} X^{\alpha} Z^{\beta} \tilde{E})[1]$ as the first component of the conjugated Pauli and the bit string

$L_{\tilde{E},\delta} = x(\tilde{E}^\dagger X^\alpha Z^\beta \tilde{E})[1:]$ as the x-coordinate of the remaining components of the conjugated Pauli. Now the state can be written as

$$\frac{1}{2^N}\sum_{\delta,\delta'}|\delta\rangle\langle\delta'|^{M,S} \otimes (P_{\tilde{E},\delta}(\rho)P^\dagger_{\tilde{E},\delta'} \otimes |L_{\tilde{E},\delta}\rangle\langle L_{\tilde{E},\delta'}| \otimes 0^t)^{C,R}$$

and (1) becomes

$$\begin{aligned}
&P^{M,S,C} P_R E'^R \pi^{C,R} \cdot \Big(\frac{1}{2^N}\sum_{\delta,\delta'}|\delta\rangle\langle\delta'|^{M,S} \otimes (P_{\tilde{E},\delta}(\rho)P^\dagger_{\tilde{E},\delta'} \otimes |L_{\tilde{E},\delta}\rangle\langle L_{\tilde{E},\delta'}| \otimes 0^t)^{C,R}\Big)\\
&= P^{M,S,C} \cdot \Big(\frac{1}{2^N}\sum_{\delta,\delta'}|\delta\rangle\langle\delta'|^{M,S} \otimes |L_{\tilde{E},\delta}\rangle\langle L_{\tilde{E},\delta'}|^C \otimes P_R E' \cdot (P_{\tilde{E},\delta}(\rho)P^\dagger_{\tilde{E},\delta'} \otimes 0^t)^R\Big)\\
&= P^{M,S,C} \cdot \Big(\frac{1}{2^N}\sum_{\delta,\delta'}|L^*_{\tilde{E}}(\delta)\rangle\langle L^*_{\tilde{E}}(\delta')|^{M,S,C} \otimes \tilde{\sigma}^R_{\delta,\delta'}\Big) \qquad (2)
\end{aligned}$$

The last equality is simplified using the abbreviation $L^*_{\tilde{E}}(\delta) = (\delta, L_{\tilde{E},\delta})$ as the concatenated string and $\tilde{\sigma}^{R,W}_{\delta,\delta'} = (P_R E' \cdot (P_{\tilde{E},\delta}(\rho)P^\dagger_{\tilde{E},\delta'} \otimes 0^t))^R \otimes \rho_{\mathsf{aux}}{}^W$ as the state in registers (R, W). Below, denote $\tilde{S}$ as the union of registers (M, S, C). Now we expand $P^{\tilde{S}} = (Z^b X^a)^{\tilde{S}}$ and decompose the attack $A^{\tilde{S},R,W} = \sum_x A_x{}^{\tilde{S},R,W}(X^{x\tilde{S}} \otimes \mathbb{1}^{R,W})$ where $A_x{}^{\tilde{S},R,W} = \sum_z Z^{z\tilde{S}} \otimes A_{x,z}{}^{R,W}$. Then from (2),

$$\begin{aligned}
&\Pi^{\tilde{S}} A^{\tilde{S},R,W} \mathsf{TP}^{M,S} \cdot (\sigma, \mathcal{S}, \mathcal{C}, \mathcal{R}, \rho_{\mathsf{aux}})\\
=&\Pi^{\tilde{S}} A^{\tilde{S},R,W} P^{\tilde{S}} \cdot \Big(\frac{1}{2^N}\sum_{\delta,\delta'}|L^*_{\tilde{E}}(\delta)\rangle\langle L^*_{\tilde{E}}(\delta')|^{\tilde{S}} \otimes \tilde{\sigma}^{R,W}_{\delta,\delta'}\Big)\\
=&\Pi^{\tilde{S}} \frac{1}{2^N}\sum_{x,x'} A_x (X^x Z^b X^a)^{\tilde{S}} \Big(\sum_{\delta,\delta'}|L^*_{\tilde{E}}(\delta)\rangle\langle L^*_{\tilde{E}}(\delta')|^{\tilde{S}} \otimes \tilde{\sigma}^{R,W}_{\delta,\delta'}\Big)(X^a Z^b X^{x'})^{\tilde{S}} A^\dagger_{x'}\\
=&\Pi^{\tilde{S}} \frac{1}{2^N}\sum_{x,x'}\sum_{\delta,\delta'}(-1)^{\langle b, x\oplus x'\rangle} Z^{b\tilde{S}} A_x \big(|L^*_{\tilde{E}}(\delta)\oplus x \oplus a\rangle\langle L^*_{\tilde{E}}(\delta')\oplus x' \oplus a)| \otimes \tilde{\sigma}_{\delta,\delta'}\big) A^\dagger_{x'} Z^{b\tilde{S}}
\end{aligned}$$

The Z^b operator takes no effect right before measurement. Moreover, if m is the measurement result on registers $\tilde{S}$ and that m passes cMPC's test, we have $m \oplus a \in \textsc{Range}(L^*_{\tilde{E}})$. Then $x = L^*_{\tilde{E}}(\delta) \oplus a \oplus m \in \textsc{Range}(L^*_{\tilde{E}})$ since $L^*_{\tilde{E}}$ is a linear function. The post-measurement state when passing cMPC's test becomes

$$\frac{1}{2^N}\sum_{y,x,x' \in \textsc{Range}(L^*_{\tilde{E}})}(-1)^{\langle b, x\oplus x'\rangle} A_x{}^{\tilde{S},R,W}\big(|y \oplus a\rangle\langle y \oplus a|^{\tilde{S}} \otimes \tilde{\sigma}^{R,W}_{\delta,\delta'}\big) A^\dagger_{x'}{}^{\tilde{S},R,W}$$

where δ, δ' are the initial N bits of $(x \oplus y), (x' \oplus y)$. Next, cMPC discards the information of P i.e., taking $\mathop{\mathbb{E}}_P[\cdot]$ on the state. We first take $\mathop{\mathbb{E}}_b[\cdot]$ to get

$$\frac{1}{2^N}\sum_{y,x \in \textsc{Range}(L^*_{\tilde{E}})} A_x{}^{\tilde{S},R,W}\big(|y \oplus a\rangle\langle y \oplus a|^{\tilde{S}} \otimes \tilde{\sigma}^{R,W}_{\delta,\delta}\big) A^\dagger_x{}^{\tilde{S},R,W}$$

Then we take $\mathop{\mathbb{E}}_a[\cdot]$ to get

$$\frac{1}{2^N}\sum_{y,x\in\textsc{Range}(L^*_{\tilde{E}})} A_x{}^{\tilde{S},R,W}\Big(\big(\frac{\mathbb{1}}{2}\big)^{\otimes(N+t)}{}^{\tilde{S}} \otimes \tilde{\sigma}_{\delta,\delta}^{R,W}\Big) A_x^{\dagger}{}^{\tilde{S},R,W}$$
$$=\mathop{\mathbb{E}}_{\delta}\Big[\sum_{x\in\textsc{Range}(L^*_{\tilde{E}})} A_x{}^{\tilde{S},R,W}\Big(\big(\frac{\mathbb{1}}{2}\big)^{\otimes(N+t)}{}^{\tilde{S}} \otimes \tilde{\sigma}_{\delta,\delta}^{R,W}\Big) A_x^{\dagger}{}^{\tilde{S},R,W}\Big]$$

Finally, cMPC discards extra information of $\tilde{E}, P_R$ except for the correction Pauli operator $P_R E'(P_{\tilde{E},(m\oplus a)[:N]}\otimes\mathbb{1}^t)E'^{\dagger} = P_R E'(P_{\tilde{E},\delta\oplus x[:N]}\otimes\mathbb{1}^t)E'^{\dagger}$, which we denote as $P'_{\tilde{E},\delta,x}$. Augmented with this last message from cMPC in O, the end state is

$$\mathop{\mathbb{E}}_{\tilde{E},P_R,\delta}\Big[\sum_{x\in\textsc{Range}(L^*_{\tilde{E}})} A_x{}^{\tilde{S},R,W}\Big(\big(\frac{\mathbb{1}}{2}\big)^{\otimes(N+t)}{}^{\tilde{S}} \otimes \tilde{\sigma}_{\delta,\delta}^{R,W}\Big) A_x^{\dagger}{}^{\tilde{S},R,W} \otimes |P'_{\tilde{E},\delta,x}\rangle\langle P'_{\tilde{E},\delta,x}|^{O}\Big]$$
$$=\Big(\sum_{x=0}\mathop{\mathbb{E}}_{\tilde{E}}(\cdots)\Big) + \Big(\sum_{x:\,x[:N]\neq 0}\mathrm{negl}(t)\mathop{\mathbb{E}}_{\tilde{E}:\,x\in\textsc{Range}(L^*_{\tilde{E}})}(\cdots)\Big) \tag{3}$$

We split it into a sum of terms corresponding to different values of x. The expression uses the fact that $0\in\textsc{Range}(L^*_{\tilde{E}})$ for every $\tilde{E}$, and that every non-zero value of x has only negligible probability over $\tilde{E}$ of lying in $\textsc{Range}(L^*_{\tilde{E}})$ (lemma 3.4 in [ABOEM17]). As the second parenthesis is negligible, the end state is indistinguishable from the first term:

$$\approx\mathop{\mathbb{E}}_{\tilde{E},P_R,\delta}\Big[A_0\Big(\big(\frac{\mathbb{1}}{2}\big)^{\otimes(N+t)}{}^{\tilde{S}} \otimes P_R E'\cdot\big((P_{\tilde{E},\delta}\cdot\rho)\otimes 0^t\big)^{R}\otimes\rho_{\mathsf{aux}}{}^{W}\Big)A_0^{\dagger}\otimes|P'_{\tilde{E},\delta,0}\rangle\langle P'_{\tilde{E},\delta,0}|^{O}\Big]$$
$$=\mathop{\mathbb{E}}_{P'\in\mathscr{P}_{1+t}}\Big[A_0\Big(\big(\frac{\mathbb{1}}{2}\big)^{\otimes(N+t)}{}^{\tilde{S}} \otimes P'E'\cdot(\rho\otimes 0^t)^{R}\otimes\rho_{\mathsf{aux}}{}^{W}\Big)A_0^{\dagger}\otimes|P'\rangle\langle P'|^{O}\Big] \tag{4}$$

On the other hand, the protocol aborts when an adversary chooses to abort cMPC, or an adversarial sender fails the test. In these cases, the quantum one-time pad P_R on register R is never handed out, so R stays maximally mixed. The end state is indistinguishable from $A\big((\frac{1}{2})^{\otimes(N+t)}{}^{\tilde{S}}\otimes(\frac{1}{2})^{\otimes(1+t)}{}^{R}\otimes\rho_{\mathsf{aux}}{}^{W}\big)A^{\dagger}$ for the former and $\sum_{x\neq 0}A_x\big((\frac{1}{2})^{\otimes(N+t)}{}^{\tilde{S}}\otimes(\frac{1}{2})^{\otimes(1+t)}{}^{R}\otimes\rho_{\mathsf{aux}}{}^{W}\big)A_x^{\dagger}$ for the latter case.

Simulating the Real World The simulator for an adversary that corrupts both the sender and the receiver is as follows. The other cases follow similarly.

Simulator 1 $\mathsf{Sim}^{\mathsf{AQA}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for AQA

1. Input: $\sigma = E\cdot(\rho\otimes 0^t)$
2. Create EPR pair $(\tilde{e}_S,\tilde{e}_R)$ of size $(N+t)$. Send $(\mathsf{TP}^{\dagger}{}^{M,S}\cdot\tilde{e}_S^{M,S,C})$ to sender.
   Create EPR pair $(\hat{e}_S,\hat{e}_R)$ of size $(1+t)$. Send $\hat{e}_R^{R}$ to receiver.

3. Receive the measurement results $r = (r_z, r_x, r_c)$ from the sender.
   Measure $\tilde{e}_R$ and compare the measurement result $\tilde{r} \leftarrow \Pi(\tilde{e}_R)$ with r.
4. Invoke ideal functionality:
   If $\tilde{r} = r$, send σ to T. Otherwise, send $\perp$ to T.
5. Output recovery:
   Upon receiving σ' from T, teleport σ' into $\hat{e}_S$.
   Send $(z, x) \leftarrow \mathsf{TP.Send}(\sigma', \hat{e}_S)$ to the receiver.
6. Output: the final output of $\mathcal{A}$.

Consider again the attack $\Pi^{\tilde{S}} A^{\tilde{S},R,W} \mathsf{TP}^{M,S}$ on $(\mathsf{TP}^{\dagger M,S} \cdot \tilde{e}_S^{M,S,C}, \hat{e}_R^R, {\rho_{\mathsf{aux}}}^W)$, where $A^{\tilde{S},R,W} = \sum_x {A_x}^{\tilde{S},R,W} X^{x\tilde{S}}$. The resulting state together with $\tilde{r}$ is

$$\Pi^{\tilde{S}}(\sum_{x,x'} {A_x}^{\tilde{S},R,W} X^{x\tilde{S}} (\tilde{e}_S^{\tilde{S}} \otimes \hat{e}_R^R \otimes {\rho_{\mathsf{aux}}}^W) X^{x'\tilde{S}} {A_{x'}^{\dagger}}^{\tilde{S},R,W}) \otimes \Pi(\tilde{e}_R)$$
$$= \mathop{\mathbb{E}}_{\tilde{r}} \big[\sum_x {A_x}^{\tilde{S},R,W} (|x+\tilde{r}\rangle\langle x+\tilde{r}|^{\tilde{S}} \otimes \hat{e}_R^R \otimes {\rho_{\mathsf{aux}}}^W) {A_x^{\dagger}}^{\tilde{S},R,W} \otimes |\tilde{r}\rangle\langle\tilde{r}|\,\big]$$

Conditioned on no abort so that $r = \tilde{r}$, the state of the adversary is

$${A_0}^{\tilde{S},R,W}\big({(\frac{\mathbb{1}}{2})^{N+t}}^{\tilde{S}} \otimes \hat{e}_R^R \otimes {\rho_{\mathsf{aux}}}^W\big) {A_0^{\dagger}}^{\tilde{S},R,W}$$

In this case, the simulator chooses to send $\sigma = E \cdot (\rho \otimes 0^t)$ to T and receives $\sigma' = E' \cdot (\rho \otimes 0^t)$ in return. After the simulator teleports σ' through $\hat{e}_S$ and sends over the measurement result (z, x) which defines the Pauli $P = X^x Z^z$, the state of the adversary becomes

$$\mathop{\mathbb{E}}_{P} \big[{A_0}^{\tilde{S},R,W}\big({(\frac{\mathbb{1}}{2})^{N+t}}^{\tilde{S}} \otimes PE' \cdot {(\rho \otimes 0^t)}^R \otimes {\rho_{\mathsf{aux}}}^W\big) {A_0^{\dagger}}^{\tilde{S},R,W} \otimes |P\rangle\langle P|^O\big]$$

which is indistinguishable from (4). When there is an abort, R holds half an EPR pair with the other half unused throughout the simulation. Thus, R is maximally mixed, and the end states are again indistinguishable from the real world.

5.3 Generalized Online Input

In protocol 1, the sender's online input is a fresh ciphertext $\sigma = E \cdot (\rho \otimes 0^t)$. In our application, the ciphertext may come from a previous teleportation result where the attacker is partly involved. To capture this case, we first introduce some notation describing quantum teleportation under attack [BGS13]. Suppose (e_S, e_R) is an EPR pair of length n, and a unitary map $A = \sum_{P_M, P_S} {P_M}^M \otimes {P_S}^S \otimes {A_{P_M,P_S}}^R$, decomposed in the Pauli basis, is applied on $(\psi^M, {e_S}^S, F({e_R}^R))$ right before the normal teleportation algorithm $\mathsf{TP.Send}$. The resulting state becomes

$$\frac{1}{2^n}\sum_{\substack{z,P_M,P_S\\x,P'_M,P'_S}} |z\rangle\langle z|^M \otimes |x\rangle\langle x|^S \otimes A_{P_M,P_S}(F(P_S^\dagger(X^xZ^z)P_M\psi P'^\dagger_M(X^xZ^z)^\dagger P'_S{}^R))A^\dagger_{P'_M,P'_S}$$

We make the following abbreviation.

$$\begin{aligned}
\Delta &:= (z, x, P_M, P_S, P'_M, P'_S)\\
\psi^{\Delta^*} &:= P_S^\dagger(X^xZ^z)P_M\psi P'^\dagger_M(X^xZ^z)^\dagger P'_S\\
\psi^{\Delta} &:= (X^xZ^z)^\dagger\psi^{\Delta^*}(X^xZ^z)\\
A^{\Delta^*}(\psi\,; F(\cdot)) &:= \frac{1}{2^n}\sum_{\Delta} |z\rangle\langle z|^M \otimes |x\rangle\langle x|^S \otimes A_{P_M,P_S}(F(\psi^{\Delta^*R}))A^\dagger_{P'_M,P'_S}\\
A^{\Delta}(\psi\,; F(\cdot)) &:= \frac{1}{2^n}\sum_{\Delta} |z\rangle\langle z|^M \otimes |x\rangle\langle x|^S \otimes A_{z,x,P_M,P_S}(F(\psi^{\Delta R}))A^\dagger_{z,x,P'_M,P'_S} \qquad (5)
\end{aligned}$$

where A_{z,x,P_M,P_S} is the conjugation of A_{P_M,P_S} by F and X^xZ^z. The notations A^{Δ^*}, A^{Δ} are useful for representing the teleportation result under attack A.

In the general case, we allow the sender's initial offline and online state to take the form $A^{\Delta}(\rho\,; (E\cdot(\cdot,0), \mathcal{S}, \mathcal{C}, \mathcal{R}, \rho_{\mathsf{aux}}))$ instead of $E\cdot(\rho, 0)$, where $\rho, \rho_{\mathsf{aux}}$ are inputs independent of the preprocessing data $(E, \mathcal{S}, \mathcal{C}, \mathcal{R})$ and A is any attack. Sections 5.1, 5.2 correspond to the case $A = \mathbb{1}$. For an arbitrary A, the security guarantee can be reformulated as indistinguishability between the following two hybrids. In hybrid 1, the simulator teleports σ to $\mathcal{A}$ and then $\mathcal{A}$, who now holds $A^{\Delta}(\rho\,; (E\cdot(\cdot,0), \mathcal{S}, \mathcal{C}, \mathcal{R}, \rho_{\mathsf{aux}}))$, runs Π^{AQA}. In hybrid 2, the simulator sends σ to the ideal functionality and simulates the adversary with $\mathsf{Sim}^{\mathsf{AQA}}$.

Hybrid 1

1. Input:
   - $\mathcal{A}$: $A^{\Delta}(\rho\,; (e_R, \mathcal{S}, \mathcal{C}, \mathcal{R}, \rho_{\mathsf{aux}}))$
   - Sim: $\sigma = E\cdot(\rho^{\Delta}\otimes 0^t), e_S$
2. Sim runs TP.Send(σ, e_S) and sends the result (z, x) to $\mathcal{A}$
3. $\mathcal{A}$ runs Π^{AQA} using cMPC

Hybrid 2

1. Input:
   - $\mathcal{A}$: $A^{\Delta}(\rho\,; (\hat{e}_R, \mathsf{TP}^\dagger\cdot\tilde{e}_S, \rho_{\mathsf{aux}}))$
   - Sim: $\sigma = E\cdot(\rho^{\Delta}\otimes 0^t), \hat{e}_S, \tilde{e}_R$
2. Sim runs $\mathsf{Sim}_{\mathcal{A}}^{\mathsf{AQA}}$ using $\mathsf{Ideal}^{\mathsf{AQA}}$

The security proof is the same as Lemma 5.1 once we write A^{Δ} back in its original form in (5). The only change is to replace ρ with ρ^{Δ}. When no abort happens, both hybrids end in the state $A'^{\Delta}(\rho\,; E'\cdot(\cdot))$ for some A' that depends only on A, $\mathcal{A}$, and the random Pauli that $\mathcal{A}$ receives at the end of the hybrids. In case of an abort, both hybrids end in the state $A'^{\Delta}(\rho\,; \mathcal{T}_{\mathscr{P}}(\cdot))$ instead.

5.4 Compatibility of Quantum Authentication Code

We have already seen AQA in Clifford-code form above. In general, our AQA can be realized under a variety of quantum authentication codes as long as the following criteria are met:

- The encoding procedure takes the form $\rho \mapsto E \cdot (\rho \otimes 0^\lambda)$ for some Clifford E.
- There is a decoding procedure that performs $E^\dagger$, measures part of the ancilla, and determines authenticity from the measurement result. Then apply a Clifford D on the remaining qubits and trace out all ancilla to recover ρ.

Requiring E, D to be Clifford gates are essential because we have to change their order with quantum teleportation and quantum one-time pad, both of which produce Pauli operators. The security proof of Lemma 5.1 are mostly independent of the underlying authentication scheme. The only place where we explicitly relied on Clifford code is in equation (3), which is exactly the property that makes Clifford code a secure quantum authentication code. One can make corresponding modifications to that part of the proof for other authentication codes. Below we demonstrate the Trap code form AQA as an example.

AQA in Trap Code Form Suppose QECC encodes 1 qubit into t qubits and corrects d errors. Trap code meets the criteria with $\lambda = 3t-1$, $E = (X^x Z^z \pi) \circ (\mathsf{QECC} \otimes \mathbb{1}^t \otimes H^t)$, and $D = \mathsf{QECC.Dec} \circ \mathsf{QECC}$. Initially, the trusted setup prepares an EPR pair (e_S, e_R) of length $3t$, pretends e_R to be a Trap code codeword, and applies $D \circ E^\dagger$ to e_R, Denote the result as (μ, α, τ), where $\mu \in \mathcal{D}^1$ is the alleged message, $\alpha \in \mathcal{D}^{t-1}$ is the alleged ancilla for QECC, and $\tau \in \mathcal{D}^{2t}$ is the alleged traps. The trusted setup authenticates μ with fresh ancilla into the receiving portal, discards α, and turns τ into the checking portal. In the online phase, classical MPC counts $u = ||x(E^\dagger X^{r'_x} Z^{r'_z} E)[t:] - r'_c||_1$ as the difference between the expected and the actual bit-flips on traps. It chooses to abort the sender if $u \geq d$; otherwise, it calculates the correction Pauli for the receiving portal. It is the conjugation of $X^{r'_x} Z^{r'_z}$ by E and D. For the security, note that some terms in equation (3) becomes non-negligible. Still, all these terms satisfy $||x||_1 < d$ by the security of Trap code. The end state of the real world in acceptance is now indistinguishable from

$$\mathop{\mathbb{E}}_{P} \Big[\sum_{||x||_1 < d} A_x{}^{\tilde{S},R,W} ((\frac{\mathbb{1}}{2})^{N+t}{}^{\tilde{S}} \otimes PE' \cdot (\rho \otimes 0^t)^R \otimes \rho_{\mathsf{aux}}{}^W) A_x^\dagger{}^{\tilde{S},R,W} \otimes |P\rangle\langle P|^O \Big]$$

which can be simulated using a similar strategy to Lemma 5.1 and [BW16].

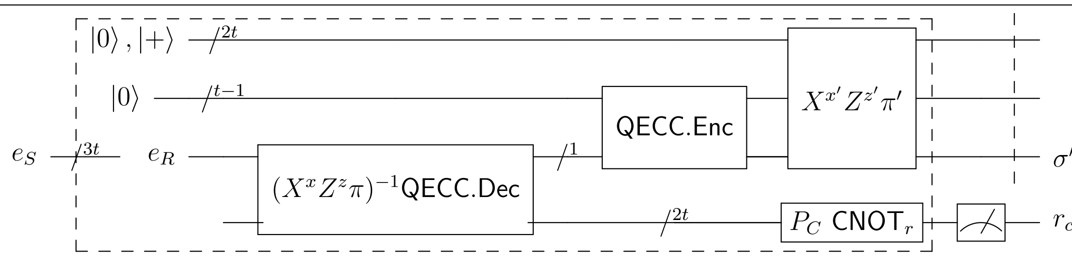

Figure 4: Auditable Quantum Authentication in Trap code form

6 BoBW-MPQC-PVIA in the Preprocessing Model

In this section, we construct a best-of-both-worlds multi-party quantum computation protocol secure with public verifiable identifiable abort (BoBW-MPQC-PVIA) from the building block AQA under a trusted setup. We break down the whole protocol into three subsections. It is useful to define the following ideal world as an intermediate step towards $\mathsf{Ideal}^{\mathsf{MPQC}}$. Roughly speaking, the ideal world of redistributed quantum computation (RQC) also computes the circuit, but keeps the result distributed and encrypted across all parties.

$\mathsf{Ideal}^{\mathsf{RQC}}$: Ideal World of Redistributed Quantum Computation

Common Input:

The security parameter 1^κ, the circuit C with input size $(\ell_1, \cdots, \ell_n)$ and total quantum output size ℓ', the number of traps $t = t(\kappa)$ on each share, and the fault-tolerant threshold $\mathtt{t}$.

Input:

P_i holds a state $\rho_i \in \mathcal{D}^{\ell_i}$.

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 samples ρ_{anc} and computes $(\rho'_1, \cdots, \rho'_n, \rho'_{\mathsf{anc}}, r') = C(\tilde{\rho}_1, \cdots, \tilde{\rho}_n; \rho_{\mathsf{anc}})$.
T encodes the output into $(s_1, \cdots, s_n) = \mathsf{QECC.Enc}(\rho'_1, \cdots, \rho'_n, \rho'_{\mathsf{anc}})$.
T samples $E_j \leftarrow \mathscr{C}_{\ell'(1+t)}$ and creates ciphertext $\sigma_j = E_j(s_j \otimes 0^{\ell' t})$.

T sends encrypted shares to malicious parties

T sends σ_j to $\mathsf{P}_j \in I$.

T receives abort messages from malicious parties

T initializes the identified corruption set $\mathtt{Corr} = \{\}$.
Each P_j in I can send $\mathtt{abort}$ message to T.
If so, T adds P_j to $\mathtt{Corr}$ and sets $E_j = \perp$.

T either aborts or sends encrypted shares to honest parties

Once $|\mathtt{Corr}| > \mathtt{t}$, T publicly aborts to all parties in $\mathtt{Corr}$.
Otherwise, T sends σ_j to $\mathsf{P}_j \notin I$.

Output:

$\mathsf{P}_1, \cdots, \mathsf{P}_n, \mathsf{O}$ output whatever they received from T.
T outputs $(E_1, \cdots, E_n)$ and r' if there is no abort.
$\mathcal{A}$ outputs its view.

6.1 Input Encoding and Ancilla Preparation

The first stage of our protocol is input encoding (IE), which transforms all inputs and proper ancilla qubits into distributed and encrypted data. One can also view IE as a simplified RQC that computes the identity circuit $C = \mathbb{1}$.

Some previous works also use QECC and authentication codes for input encoding, but let us look closer into what is missing in their procedures. On the one hand, MPQC with an honest majority [BOCG+06] only authenticates states under user-sampled keys, so the malicious parties have more control over their information. On the other hand, MPQC with a dishonest majority [ACC+21] allows users to submit incorrect QECC codewords without consequences, which hinders transversal computation on the alleged codewords. Here, we can avoid these two issues at the same time.

Our preprocessing protocol for IE work as follows. For simplicity, we will use a QECC scheme that encodes every qubit into n qubits. The trusted party performs QECC encoding, quantum authentication under secret keys, and quantum one-time pad on half EPR pair. Together with the other half EPR pair, they form the sending and receiving portals $\mathcal{S}_i, \mathcal{R}_i$, which are given to the respective parties. In the online phase, party P_i can teleport its input ρ_i through the sending portal $\mathcal{S}_i$ to fill in the contents of the receiving portals.

Protocol 2 ($\Sigma^{\mathsf{IE}}, \Pi^{\mathsf{IE}}$) for Input Encoding

---

Parameters: $n, t, (\ell_1, \cdots, \ell_n), \ell_{\mathsf{anc}}, \ell'$

Preprocessing Σ^{IE}:

1. For each $i \in [n]$, generate coordinate-wise EPR pair $(e_{S,i}, e_{R,i})$ of length ℓ_i.
   Compute $(f_{i,1}, \cdots, f_{i,n}) \leftarrow \mathsf{QECC.Enc}(e_{R,i})$.
2. Create auxiliary state $\phi_h \in \mathcal{D}^{\ell_{\mathsf{anc}}}$ consisting of ancilla and magic states.
   Compute $(f_1^{\mathsf{anc}}, \cdots, f_n^{\mathsf{anc}}) \leftarrow \mathsf{QECC.Enc}(\phi_h)$.
3. Sample random Pauli $P_j \leftarrow \mathscr{P}_{\ell(1+t)}$ and random Clifford $E_j \leftarrow \mathscr{C}_{\ell(1+t)}$.
   Compute $\mathcal{R}_j = P_j E_j \cdot ((f_{1,j} \otimes 0^{\ell_1 t}) \otimes \cdots \otimes (f_{n,j} \otimes 0^{\ell_n t}) \otimes (f_j^{\mathsf{anc}} \otimes 0^{\ell_{\mathsf{anc}} t}))$.
4. Send to P_i the sending portal $\mathcal{S}_i = e_{S,i}$ and the receiving portal $\mathcal{R}_i$.
5. Send to cMPC the secret $\{P_j, E_j\}_{j\in[n]}$.

Online Input:

1. Each P_i holds private input $\rho_i \in \mathcal{D}^{\ell_i}$.

Protocol Π^{IE}:

1. P teleports ρ_i via sending portal $\mathcal{S}_i$.
   P then sends $(z_i, x_i) \leftarrow \mathsf{TP.Send}(\rho_i, \mathcal{S}_i)$ to cMPC.
2. cMPC computes $P'_{i,1} \otimes \cdots \otimes P'_{i,n} = \mathsf{QECC}(X^{x_i} Z^{z_i} \otimes \mathbb{1}^{(n-1)\ell_i})\mathsf{QECC}^\dagger$.
   cMPC computes $P''_j = P_j E_j \big((P'_{1,j} \otimes \mathbb{1}^{\ell_1 t}) \otimes \cdots \otimes (P'_{n,j} \otimes \mathbb{1}^{\ell_n t}) \otimes \mathbb{1}^{\ell_{\mathsf{aux}}(1+t)} \big) E_j^\dagger$.
   cMPC sends the swapped out correction Pauli P''_j to P_j for $j \in [n]$.

Output:

1. P_j outputs $\sigma_j = P''^\dagger_j \mathcal{R}_j$ for $j \in [n]$.
2. cMPC outputs $\{E_j\}_{j\in[n]}$.

The input encoding protocol realizes the task of redistributed quantum computation for the identity circuit $C = \mathbb{1}$. Below we analyze the end state of the real protocol, and postpone its simulation to the next section.

Lemma 6.1. *For every non-uniform* (QPT) *adversary $\mathcal{A}_I$ corrupting I, there is a non-uniform* (QPT) *simulator $\mathsf{Sim}_{\mathcal{A}_I}$ corrupting I, such that for any states $\rho_1, \cdots, \rho_n, \rho_{\mathsf{aux}}$,*

$$\{\mathsf{Real}^{\Pi^{\mathsf{IE}}\circ\Sigma^{\mathsf{IE}}}_{\mathcal{A}(\rho_{\mathsf{aux}})}(1^\kappa, \rho_1, \cdots, \rho_n)\} \approx \{\mathsf{Ideal}^{\mathsf{RQC}}_{\mathsf{Sim}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}}(1^\kappa, \mathbb{1}, \rho_1, \cdots, \rho_n)\} \tag{6}$$

It suffices to consider attacks on the protocol that happen in protocol step 1. Say registers (S_i, R_j, I_i, W, O_j) store $(\mathcal{S}_i, \mathcal{R}_j, \rho_i, \rho_{\mathsf{aux}}, P''_j)$ respectively. We can always rewrite the attack as $(\mathsf{TP.Send} \circ A)$ for some unitary map A. Using the abbreviation of (5), the end state of the real protocol is

$$\begin{aligned} &A^{\Delta^*}\Big(\bigcup_{i\in[n]}\rho_i\ ;\ (\bigotimes_{j\in[n]} P_j E_j)(\mathsf{QECC.Enc}(\cdot, \phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W}\Big) \\ =&A^{\Delta}\Big(\bigcup_{i\in[n]}\rho_i\ ;\ (\bigotimes_{j\in[n]} P''_j E_j)(\mathsf{QECC.Enc}(\cdot, \phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W}\Big) \end{aligned} \tag{7}$$

The real protocol can be simulated with the following simulator.

Simulator 2 $\mathsf{Sim}^{\mathsf{IE}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for Input Encoding

1. Create EPR pair $(\tilde{e}^{S_i}_{S_i}, \tilde{e}^{R'_i}_{R,i})$ of size ℓ_i.
   Send $\tilde{e}_S$ in register S_i to party P_i for $i \in [n]$.
2. Create EPR pair $(\hat{e}_{S,j}, \hat{e}^{R_j}_{R,j})$ of size $\ell(1+t)$, and sample $\hat{P}_j \leftarrow \mathscr{P}_{\ell(1+t)}$.
   Send $\hat{P}_j \cdot \hat{e}_{R,j}$ in register R_j to party P_j for $j \in [n]$.
3. On input $\rho_i \in \mathcal{D}^{\ell_i}$, send ρ_i in register I_i to party P_i.
   Receive the teleportation measurement result (z_i, x_i) from party P_i.
   Obtain the teleported input $\rho'_i \leftarrow \mathsf{TP.Receive}(z_i, x_i, \tilde{e}_{R,i})$.
4. Invoke ideal functionality:
   Send ρ'_i to T. Do not send any `abort` message to T. Receive σ_j from T.
5. Teleport σ_j into $\hat{e}_{S,j}$, resulting in the outcomes $(z'_j, x'_j) \leftarrow \mathsf{TP.Send}(\sigma_j, \hat{e}_{S,j})$.
   Send $P''_j = \hat{P}_j X^{x'_j} Z^{z'_j}$ to party P_j.
6. Output the final output of $\mathcal{A}$.

Let R'_i denote the register holding $\tilde{e}_{R,i}$. Using notation (5), simulation step 3 ends with

$$\sum_{\Delta}\Big(|z_i\rangle\langle z_i|^{I_i}, |x_i\rangle\langle x_i|^{S_i}, \bigcup_{i\in[n]}\rho_i^{\Delta\,R'_i}, {U_{P_{I_i},P_{S_i}}}^{R_j,W}({\hat{P}_j \cdot \hat{e}_{R,j}}^{R_j}, {\rho_{\mathsf{aux}}}^{W}){U^\dagger_{P'_{I_i},P_{S'_i}}}^{R_j,W}\Big)$$

After steps 4, 5, the values in registers R_i' is correctly QECC encoded, securely authenticated, and being teleported. The end state of the simulation becomes

$$\begin{aligned} &A^{\Delta}\Big(\bigcup_{i\in[n]} \rho_i \ ; \ \big(\bigotimes_{j\in[n]} \hat{P}_j X^{x_j'} Z^{z_j'} E_j\big)(\mathsf{QECC.Enc}(\cdot, \phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W} \Big) \\ =&A^{\Delta}\Big(\bigcup_{i\in[n]} \rho_i \ ; \ \big(\bigotimes_{j\in[n]} P_j'' E_j\big)(\mathsf{QECC.Enc}(\cdot, \phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W} \Big) \end{aligned}$$

which is the same as the real world state. Since the adversary learns P_j'' in the end, we can also rewrite the state as

$$A'^{\Delta}\Big(\bigcup_{i\in[n]} \rho_i \ ; \ \big(\bigotimes_{j\in[n]} E_j\big)(\mathsf{QECC.Enc}(\cdot, \phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W} \Big)$$

for some A' that depends on the random Pauli operators P_j''.

6.2 Distributed Circuit Evaluation

After spreading out QECC codewords of the inputs across parties, the players proceed to evaluate a circuit C in a distributed manner. As long as there are enough additional ancilla and magic states, C can be computed using only certain Clifford gates and measurements. To come up with a protocol for this task, we combine ideas from [DGJ+20] with transversal computation over QECC codewords.

Protocol 3 $(\Sigma^{\mathsf{RQC}}, \Pi^{\mathsf{RQC}})$ for Redistributed Quantum Computation

---

Preprocessing: $\Sigma^{\mathsf{RQC}} = \Sigma^{\mathsf{IE}}$
Online Input:

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

Protocol Π^{RQC}**:**

1. Parties run Π^{IE}. cMPC initializes the set $\texttt{Corr} = \{\}$.
   At any time if $|\texttt{Corr}| > \mathsf{t}$, cMPC publicly aborts to all players in $\texttt{Corr}$.
2. Iteratively, cMPC decides whether a Clifford gate G has to be applied on the joint logical state, or some logical qubit has to go through a measurement.
3. To perform Clifford gate G on the l logical qubits, do for every $\mathsf{P}_j \notin \texttt{Corr}$:
   (a) Suppose ciphertext σ_j held by party P_j is authenticated under key E_j.
   (b) cMPC updates the key with $E_j' = E_j(G \otimes \mathbb{1}^{lt})^{\dagger}$.
4. To perform a measurement on 1 of l logical qubits, do for every $\mathsf{P}_j \notin \texttt{Corr}$:
   (a) Suppose ciphertext σ_j held by party P_j is authenticated under key E_j.

(b) cMPC samples strings $c_j^z, c_j^x, c_j^t \in \{0,1\}^{1+t}$ and key $E'_j \in \mathscr{C}_{(l-1)(1+t)}$.
cMPC sends $V_j = (E'_j \otimes Z^{c_j^z} X^{c_j^x} \mathsf{CNOT}_{c_j^t}) E_j^\dagger$ to P_j.

(c) P_j calculates and parses $(\sigma'_j, \tau'_j) = V_j \cdot \sigma_j$ where $\tau'_j \in \mathcal{D}^{1+t}$.
P_j measures τ'_j and sends the outcome $r_j \leftarrow \Pi(\tau'_j)$ to cMPC.

(d) cMPC sets bit b_j if $r_j \oplus c_j^x = b_j(1, c_j^t)$ has a solution.
Otherwise, set $b_j = \perp$ and add P_j into `Corr`.
cMPC computes $b \leftarrow \mathsf{QECC.Dec}(b_1, \cdots, b_n)$ as the measurement result.

(e) P_j now holds ciphertext σ'_j authenticated under key E'_j.

Output:

1. P_j outputs the ciphertext it currently holds.
2. cMPC outputs the authentication key of each party (set to $\perp$ if party $\in$ `Corr`).

Our protocol $(\Sigma^{\mathsf{RQC}}, \Pi^{\mathsf{RQC}})$ can perform redistributed quantum computation as specified by $\mathsf{Ideal}^{\mathsf{RQC}}$ on any circuit. This is captured by the following lemma.

Lemma 6.2. *For every non-uniform (QPT) adversary $\mathcal{A}_I$ corrupting I, there is a non-uniform (QPT) simulator $\mathsf{Sim}_{\mathcal{A}_I}$ corrupting I, such that for any states $\rho_1, \cdots, \rho_n, \rho_{\mathsf{aux}}$,*

$$\{\mathsf{Real}^{\Pi^{\mathsf{RQC}} \circ \Sigma^{\mathsf{RQC}}}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}(1^\kappa, C, \rho_1, \cdots, \rho_n)\} \approx \{\mathsf{Ideal}^{\mathsf{RQC}}_{\mathsf{Sim}_{\mathcal{A}_I(\rho_{\mathsf{aux}})}}(1^\kappa, C, \rho_1, \cdots, \rho_n)\} \tag{8}$$

Proof. First, we construct the simulator as follows.

Simulator 3 $\mathsf{Sim}^{\mathsf{RQC}}_{\mathcal{A}(\rho_{\mathsf{aux}})}$ for Redistributed Quantum Computation

---

1. Generate preprocessing data:
   - Create EPR pair $(\tilde{e}_{S_i}^{S_i}, \tilde{e}_{R,i}^{R'_i})$ of size ℓ_i for input encoding.
   - Create EPR pair $(\hat{e}_{S,j}^{(0)}, \hat{e}_{R,j}^{(0)})$ of size $\ell'(1+t)$ for output recovery.
   - Create EPR pair $(\hat{e}_{S,j}^{(l)}, \hat{e}_{R,j}^{(l)})$ of size $(1+t)$ for measurement.
   - Sample $\hat{P}_j \leftarrow \mathscr{P}_{\ell(1+t)}$ and $\hat{E}_j^{(l)} \leftarrow \mathscr{C}_{1+t}$ for $l \in \{\ell'+1, \cdots, \ell\}$.
   - Send $\tilde{e}_{S_i}$ in register S_i to P_i.
   - Send $\hat{P}_j \cdot \big(\hat{e}_{R,j}^{(0)}, (\hat{E}_j^{(\ell'+1)} \cdot \hat{e}_{R,j}^{(\ell'+1)}), \cdots, (\hat{E}_j^{(\ell)} \cdot \hat{e}_{R,j}^{(\ell)})\big)$ in register R_j to P_j.
2. Input extraction:
   - On input $\rho_i \in \mathcal{D}^l$, send ρ_i in register I_i to party P_i.
   - Receive the teleportation measurement result (z_i, x_i) from party P_i.
   - Extract the input $\rho'_i \leftarrow \mathsf{TP.Receive}(z_i, x_i, \tilde{e}_{R,i})$ and send it to T.

3. Output recovery:
   - Upon receiving σ_j from T, teleport σ_j into the first component $\hat{e}^{(0)}_{S,j}$.
   - Obtain $(z'_j, x'_j) \leftarrow \mathsf{TP.Send}(\sigma_j, \hat{e}^{(0)}_{S,j})$.
   - Send $P''_j = \hat{P}_j(X^{x'_j}Z^{z'_j} \otimes \mathbb{1}^{(\ell-\ell')(1+t)})$ to P_j.
4. For each measurement step of C, do the following (with l from ℓ to $\ell'+1$):
   - Send to P_j the gate $\hat{E}^{(l)}_j$.
   - Receive the measurement result $r_j \in \{0,1\}^{1+t}$ from party P_j.
   - Measure $\hat{e}^{(l)}_{S,j}$ and compare the result $\tilde{r}_j = \Pi(\hat{e}^{(l)}_{S,j})$ with r_j.
   - If $\tilde{r}_j \neq r_j$, let P_j send `abort` to T.

Output: the final output of $\mathcal{A}$.

We prove by induction on the number of circuit steps $|C|$ that the real protocol can be simulated by the simulator $\mathsf{Sim}^{\mathsf{RQC}}$. We also prove in the induction that the end state of both the real and ideal world when P_j is not being marked as corrupted are indistinguishable from a state where cMPC holds random Clifford keys $\{E_j\}_{j\in[n]}$ and the adversary holds

$$A^{\Delta}\big(\bigcup_{i\in[n]} \rho_i \ ; \ (\bigotimes_{j\in[n]} E_j)(\mathsf{QECC.Enc}(C(\cdot,\phi_h)), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W}\big) \tag{9}$$

for some map A. For the base case $C = \mathbb{1}$, the protocol is equal to $(\Sigma^{\mathsf{IE}}, \Pi^{\mathsf{IE}})$. From the analysis of the previous section, it remains to show that $\mathsf{Sim}^{\mathsf{RQC}}(\mathbb{1})$ ends in state (7). Using notation (5), the state at the end of simulator step 2 is

$$\sum_{\Delta} \big(|z_i\rangle\langle z_i|^{I_i}, |x_i\rangle\langle x_i|^{S_i}, \bigcup_{i\in[n]} {\rho_i^{\Delta}}^{R'_i}, {U_{P_{I_i},P_{S_i}}}^{R_j,W}(\hat{P}_j \cdot {\hat{e}^{(0)}_{R,j}}^{R_j}, {\rho_{\mathsf{aux}}}^{W}){U^{\dagger}_{P'_{I_i},P_{S'_i}}}^{R_j,W}\big)$$

After step 3, the values in registers R'_i is correctly QECC-encoded, securely authenticated, and being teleported. The end state of the simulation becomes

$$\begin{aligned}
&A^{\Delta}\big(\bigcup_{i\in[n]} \rho_i \ ; \ (\bigotimes_{j\in[n]} \hat{P}_j X^{x'_j} Z^{z'_j} E_j)(\mathsf{QECC.Enc}(\cdot,\phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W}\big)\\
=&A^{\Delta}\big(\bigcup_{i\in[n]} \rho_i \ ; \ (\bigotimes_{j\in[n]} P''_j E_j)(\mathsf{QECC.Enc}(\cdot,\phi_h), 0^{lnt})^{R_j}, {\rho_{\mathsf{aux}}}^{W}\big)
\end{aligned}$$

which is equal to (7). (When $C = \mathbb{1}$, the simulator does nothing in step 4.)

Suppose the parties are going to run a further computation step under our induction hypothesis. For the protocol of performing a Clifford gate G, cMPC updates the keys to $\{E'_j = E_j(G\otimes\mathbb{1}^{lt})^{\dagger}\}_{j\in[n]}$. These keys are still random Clifford keys. We can rewrite the state of the real-protocol adversary in terms of $\{E'_j\}_{j\in[n]}$ as

$$\begin{aligned} &A^{\Delta}\big(\underset{i\in[n]}{\cup}\rho_i\ ;\ (\underset{j\in[n]}{\otimes}E'_j(G\otimes\mathbb{1}^{lt}))(\mathsf{QECC.Enc}(C(\cdot,\phi_h)),0^{lnt})^{R_j},\rho_{\mathsf{aux}}{}^{W}\big)\\ =&A^{\Delta}\big(\underset{i\in[n]}{\cup}\rho_i\ ;\ (\underset{j\in[n]}{\otimes}E'_j)(\mathsf{QECC.Enc}(G\circ C(\cdot,\phi_h)),0^{lnt})^{R_j},\rho_{\mathsf{aux}}{}^{W}\big) \end{aligned} \tag{10}$$

since the QECC supports transversal computation on these Clifford gates. As the only difference between $\mathsf{Sim}^{\mathsf{RQC}}(C)$ and $\mathsf{Sim}^{\mathsf{RQC}}(G\circ C)$ is the ideal functionalities they interact with, we see that $\mathsf{Sim}^{\mathsf{RQC}}(G\circ C)$ also ends with state (10).

For the protocol of measuring a logical qubit shared over registers $Q_j\subseteq R_j$, the adversary receives $V_j=(E'_j\otimes Z^{c^z_j}X^{c^x_j}\mathsf{CNOT}_{c^t_j})E^\dagger_j$, so its state can be described as

$$A'^{\Delta}\big(\underset{i\in[n]}{\cup}\rho_i\ ;\ (\underset{j\in[n]}{\otimes}(E'_j\otimes Z^{c^z_j}X^{c^x_j}\mathsf{CNOT}_{c^t_j}))(\mathsf{QECC.Enc}(C(\cdot,\phi_h)),0^{lnt})^{R_j},\rho_{\mathsf{aux}}{}^{W}\big)$$

for some A' that can depend on V_j. Next, the adversary measures Q_j. To analyze cMPC's examination, we decompose the attack A' according to Pauli basis on registers Q_j. That is, we write $A'=\Sigma_{x_j}A'_{x_j}{}^{R_j,W}X^{x_j\,Q_j}$ where $A'_{x_j}=\Sigma_{z_j}A'_{x_j,z_j}{}^{R_j\setminus Q_j,W}Z^{z_j\,Q_j}$. Similar to [DGJ+20], all Pauli P with $x(P)\neq 0$ would fail cMPC's test with overwhelming probability over random (c^z_j,c^x_j,c^t_j). Thus, the residual state passing the test would be indistinguishable from

$$A_0'^{\Delta}\big(\underset{i\in[n]}{\cup}\rho_i\ ;\ (\underset{j\in[n]}{\otimes}E'_j)(\mathsf{QECC.Enc}(C(\cdot,\phi_h))_{[:l-1]},0^{(l-1)nt})^{R_j\setminus Q_j},\rho_{\mathsf{aux}}{}^{W}\big) \tag{11}$$

while the string r_j provided by P_j is $(\Pi(\mathsf{QECC.Enc}(C(\cup_i\rho_i,\phi_h)_{[l]})_{[j]})(1,c^t_j)\oplus x_j)$. Then the bit b recovered by cMPC is $\Pi(C(\cup_i\rho_i,\phi_h)[l])$ because measurement is transversal on our QECC. Now we turn to the simulated end state. There are two main differences between $\mathsf{Sim}^{\mathsf{RQC}}(C)$ and $\mathsf{Sim}^{\mathsf{RQC}}((\mathbb{1}\otimes\Pi)\circ C)$. First, the latter simulator encrypts part of the receiving portals under key $\hat{E}^{(l)}_j$. Second, the latter simulator performs an additional measurement step. As neither simulators give $\hat{e}^{(l)}_{S,j}$ to the adversary, both $\hat{e}^{(l)}_{R,j}$ and $\hat{E}^{(l)}_j\cdot\hat{e}^{(l)}_{R,j}$ are maximally mixed states and are indistinguishable from the view of $\mathcal{A}$. Next, we decompose $A'=\Sigma_{x_j}A'_{x_j}{}^{R_j,W}X^{x_j\,Q_j}$. The only possible Pauli operators on Q_j that can yield equal measurements on $\hat{e}^{(l)}_{S,j}$ and $\hat{e}^{(l)}_{R,j}$ are those without any bit-flips. Hence, conditioned that the simulator does not make P_j send `abort`, the attack reduces to A'_0, and the end state is equal to (11).

Finally, we analyze the case when some corruptions are detected. If party P_j is identified as corrupted, the output key E_j will be replaced with $\perp$ in both the real and simulated worlds, so that the j-th share in (9) becomes the maximally mixed state in both worlds. When P_j is added into the list `Corr` because the measurement step failed, one easily sees that both worlds have residual states (indistinguishably) described by $(A'^{\Delta}-A_0'^{\Delta})(\cdot)$.

□

6.3 Multiparty Quantum Computation

The final step is to transit from RQC to MPQC. After the computation, the evaluators must send the encrypted and encoded outputs back to their owners. Apparently, the outputs could turn out ruined when the owner unpacks the messages. This situation is where AQA comes into play, which guarantees public verifiable identifiable abort when some sender modifies the message. If no such problem happens, the parties have accomplished their MPQC task. Our MPQC protocol will be a direct concatenation of RQC and AQA.

Protocol 4 $(\Sigma^{\mathsf{BoBW}_0}, \Pi^{\mathsf{BoBW}_0})$ for the Preprocessing Best-of-Both-Worlds Multiparty Quantum Computation with Public Verifiable Identifiable Abort

---

Parameters: $n, t, \mathsf{t}, (\ell_1, \cdots, \ell_n), \ell_{\mathsf{anc}}, \ell'$

Preprocessing Σ^{BoBW_0}**:**

1. Run Σ^{RQC}. Then run $\Sigma^{\mathsf{AQA}}(\mathsf{P}_j, \mathsf{P}_i)$ for every pair of parties $i, j \in [n]$.

Online Input:

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

Protocol Π^{BoBW_0}**:**

1. All parties run the protocol Π^{RQC} to compute C.
   At any time if $|\mathtt{Corr}| > \mathsf{t}$, cMPC publicly aborts to all players in $\mathtt{Corr}$.
2. Suppose σ_j that P_j obtains from step 1 is authenticated under key $E_j \neq \perp$.
   cMPC retrieves key pair $(E'_{i,j}, E''_{i,j})$ that was given by $\Sigma^{\mathsf{AQA}}(\mathsf{P}_j, \mathsf{P}_i)$.
   cMPC sends $V_j = (E'_{1,j} \otimes \cdots \otimes E'_{1,n})E_j^\dagger$ to P_j.
   P_j calculates $V_j \cdot \sigma_j$ and parses the result into $(\sigma'_{1,j}, \cdots, \sigma'_{n,j})$.
3. AQA from party P_j to party P_i: run $\Pi^{\mathsf{AQA}}(\mathsf{P}_j(\sigma'_{i,j}), \mathsf{P}_i, \mathsf{cMPC})$ for $i, j \in [n]$.
   P_i collects what it received $\{\sigma''_{i,j}\}_{j\in[n]}$ from the other parties.
   If AQA identifies P_j as malicious, put P_j into $\mathtt{Corr}$.
4. cMPC sends keys $\{E''_{i,j}\}_{j\in[n]}$ to P_i and publicly outputs classical output r'.

Output:

1. P_i outputs $\rho'_i \leftarrow \mathsf{QECC.Dec}\big(E''^{\dagger}_{i,1} \cdot \sigma''_{i,1}, \cdots, E''^{\dagger}_{i,n} \cdot \sigma''_{i,n}\big)$ and r'.

Theorem 6.3. $(\Sigma^{\mathsf{BoBW}_0}, \Pi^{\mathsf{BoBW}_0})$ *is a best-of-both-worlds multi-party quantum computation secure with public verifiable identifiable abort of threshold* t *in the preprocessing MPC-hybrid model as defined in Definition 4.1.*

Proof. In short, the simulator runs a concatenation of $\mathsf{Sim}^{\mathsf{RQC}}$ with $\mathsf{Sim}^{\mathsf{AQA}}$. $\mathsf{Ideal}^{\mathsf{MPQC}}$ is invoked in $\mathsf{Sim}^{\mathsf{RQC}}$ to get $\{\rho'_i\}_{i\in I}$. In the output recovery phase of $\mathsf{Sim}^{\mathsf{RQC}}$, the simulator teleports $\mathsf{TP}^\dagger \cdot \tilde{e}_S$ prepared by $\mathsf{Sim}^{\mathsf{AQA}}$ instead of $\hat{e}_S^{(0)}$. Whenever $|\mathtt{Corr}| > \mathtt{t}$, the simulator sends `abort` message from parties in $|\mathtt{Corr}|$ to $\mathsf{Ideal}^{\mathsf{MPQC}}$. In the output recovery phase of $\mathsf{Sim}^{\mathsf{AQA}}$, the simulator authenticates ρ'_i with fresh random keys, teleports the result to P_i, and sends the key out.

When $|I| \leq \mathtt{t}$, the protocol never reaches the abort condition. By Lemma 6.2 and Lemma 5.1, the honest parties will certainly receive at least a proportion $1-\frac{\mathtt{t}}{n}$ of their output codewords. Our requirement on QECC then implies that the honest parties can recover their outputs. From Lemma 6.2 and Lemma 5.1, we deduce that the protocol is indistinguishable from simulation using $\mathsf{Ideal}^{\mathsf{MPQC-Full}}$ when $|I| \leq \mathtt{t}$ and using $\mathsf{Ideal}^{\mathsf{MPQC}}$ when $|I| \leq n-1-\mathtt{t}$. □

7 BoBW-MPQC-PVIA without Trusted Setup

So far, we have shown an MPQC-PVIA protocol in the preprocessing model. This section presents how we discard the trusted setup and obtain an MPQC-PVIA protocol without preprocessing. The idea here is to instantiate the preprocessing phase with MPQC-SWIA e.g., [ACC+21].

Hierarchical MPQC-SWIA We quickly recall the SWIA property of [ACC+21]. When the [ACC+21] protocol aborts, their classical MPC outputs a partition of all parties into two mutually distrusting groups. By observation, one of the two groups comprises only malicious players, and the other group contains all honest players, with some malicious players pretending to be honest. We can take advantage of this information to get an MPQC-PVIA protocol. First, we replace the trusted setup of our preprocessing MPQC-PVIA by running MPQC-SWIA, where it can create preprocessing data (including portals, magic states, and ancilla) for each party. Then, if the protocol fails, we allow those two groups to run their MPQC-SWIA protocols independently. The parties will ultimately obtain reliable preprocessing data within subgroups by iterating this procedure. Note that we do not publicly identify anyone during this stage.

After generating enough reliable preprocessing data, every party can run the protocol Π^{BoBW_0} within its group to obtain the output. Since the honest parties are always in the same group, they will jointly evaluate their outputs in the Π^{BoBW_0}. 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_0}, which only aborts dishonest members.

We emphasize that the parties may divide into groups when the protocol executes successfully. In this situation, if the circuit of interest involves public output, the legitimacy of each group must be determined, and the public output selected accordingly. Our protocol does not attempt to address this. We take a step back and consider only circuits with private outputs. At the end of our protocol, either a public verifiable identifiable abort occurs, or each party is satisfied with its private outcome. Dealing with circuits with public outputs remains an intriguing open problem for one who would like to achieve a stronger PVIA notion.

Now, we formulate our protocol. Let Π^{SWIA} be [ACC+21]'s MPQC-SWIA protocol.

Protocol 5 Π^{BoBW_1} for the Best-of-Both-Worlds MPQC-PVIA

Input:

1. Everyone holds the circuit description C which has only private outputs.
2. Party P_i holds private input $\rho_i \in \mathcal{D}^{\ell_i}$.

Protocol:

1. Put $G = \{\{\mathsf{P}_1, \cdots, \mathsf{P}_n\}\}$ as the initial partition of parties (i.e., no partition). Mark the set $\{\mathsf{P}_1, \cdots, \mathsf{P}_n\} \in G$ as unfinished.
2. Repeat the following until all sets in G are marked as finished: For unfinished $S \in G$, the parties in S run Π^{SWIA} together on the circuit $\Sigma^{\mathsf{BoBW}_0}(C_S)$, where C_S is the circuit that prepares default inputs for parties $\notin S$ and runs C.
   (a) If the protocol succeeds, parties in S obtain the required preprocessing data. In this case, mark S as finished.
   (b) Otherwise, the MPQC-SWIA protocol outputs a partition $\{S_0, S_1\}$ of S. In this case, replace S with the unfinished S_0 and S_1 in partition G.
3. Run Π^{BoBW_0} within every set $S \in G$. Abort to all identified parties if any execution of Π^{BoBW_0} aborts.

Output:

1. Upon abort, all parties output the identities collected from all Π^{BoBW_0}.
2. Otherwise, P_i outputs the result obtained from his/her execution of Π^{BoBW_0}.

Theorem 7.1. Π^{BoBW_1} *is a best-of-both-worlds multi-party quantum computation secure with public verifiable identifiable abort over circuits with only private outputs in the MPC-hybrid model as defined in Definition 4.1.*

Proof. Since our protocol makes only sequential calls to MPQC-SWIA protocols, the security of MPQC-SWIA allows one to replace these protocol executions with the ideal functionality of MPQC-SWIA. The resulting hybrid world $\mathcal{H}_1$ is indistinguishable from the real world. The ideal functionality of MPQC-SWIA in $\mathcal{H}_1$ translates to the following game. At the beginning of $\mathcal{H}_1$, the adversary chooses how to partition all parties step by step, but cannot partition the honest parties. Afterward, there is a trusted setup on Σ^{BoBW_0} followed by Π^{BoBW_0} for each group of the partition. Since these preprocessing BoBW protocols are independent instances, Theorem 6.3 implies that they can be replaced with the ideal world of BoBW-MPQC-PVIA.

It is easy to see that the resulting hybrid $\mathcal{H}_2$ preserves the PVIA property. To argue full security when $|I| \leq \mathsf{t}$, one observes that the group containing all the honest parties has at least a $1 - \frac{\mathsf{t}}{n}$ proportion of honest parties, so the full security of BoBW-MPQC-PVIA is also preserved. □

8 Extentions

8.1 Constant Round MPQC-PVIA in the Preprocessing Model

When aiming only for MPQC-PVIA, our BoBW-MPQC-PVIA protocol in the preprocessing model can be made round efficient. The reason is that we no longer need to keep the data distributed across parties, which eliminates the need for information exchange during computation. Instead, a designated party can hold all the data, and the entire process can become constant-round using quantum garbled circuits for computation as in [BCKM21].

Theorem 8.1. *There is a constant-round multi-party quantum computation secure with public verifiable identifiable abort in the preprocessing MPC-hybrid model.*

Protocol 6 ($\Sigma^{\mathsf{ConstMPQC}}, \Pi^{\mathsf{ConstMPQC}}$) for the Preprocessing Constant-Round MPQC-PVIA

Preprocessing $\Sigma^{\mathsf{ConstMPQC}}$:

1. Run Σ^{IE}. Then run $\Sigma^{\mathsf{AQA}}(\mathsf{P}_i, \mathsf{P}_1)$ for every party $i \in [n]$.

Protocol $\Pi^{\mathsf{ConstMPQC}}$:

1. All parties run the input encoding protocol Π^{IE}.
   The designated evaluator P_1 obtains ciphertext σ under key E.
2. cMPC generates quantum garbled circuit:
   (a) Retrieve key pair (E_i', E_i'') that was created by $\Sigma^{\mathsf{AQA}}(\mathsf{P}_i, \mathsf{P}_1)$.
   (b) Define the augmented circuit C' as:
      - Measure all the traps. If any trap is non-zero, end by outputting $\perp$.
      - Evaluate C on the inputs and ancilla states, and apply $E_1' \otimes \cdots \otimes E_n'$.

   (c) Run $(\hat{E}, \hat{C}) \leftarrow \mathsf{QG.Garble}(C')$.
   (d) Send the gate $V = \hat{E}E^\dagger$ and the garbled circuit $\hat{C}$ to the evaluator P_1.
3. P_1 evaluates the garbled circuit $\sigma' \leftarrow \mathsf{QG.Eval}(V \cdot \sigma, \hat{C})$.
   P_1 parses the result into $(\sigma_1', \cdots, \sigma_n')$.
4. AQA from party P_1 to party P_i for $i \in [n]$: run $\Pi^{\mathsf{AQA}}(\mathsf{P}_1(\sigma_i'), \mathsf{P}_i, \mathsf{cMPC})$.
   If any Π^{AQA} fails, cMPC publicly aborts to P_1.
   Otherwise, P_i receives the ciphertext σ_i'' successfully.
5. cMPC sends key E_i'' to P_i.

Output: P_i outputs $\rho_i' \leftarrow \mathsf{C.Dec}_{E_i''}(\sigma_i'')$.

The security follows from adapting the proof of Theorem 6.3 with the proofs [BCKM21, BY22] that use garbled circuits to perform the computation.

8.2 MPQC-PVIA from Trap-code AQA

We also discover the possibility of constructing MPQC-PVIA using Trap-code form AQA (Figure 4). Existing MPQC protocols against a dishonest majority [DGJ+20, ACC+21, BCKM21] only consider Clifford authentication code because of the obstruction of the Trap authentication code's checking step. For example, if we replace the Clifford code in [ACC+21] with Trap code, the player will not know if the errors are caused by the previous player or other prior players. This is because trap code requires a permutation to hide the input qubits' information. By only considering the Trap authentication code, one cannot guarantee the secrecy of the qubits' location while still identifying the suspect.

Thanks to our AQA, by slightly twisting Trap-code and plugging it into our AQA, we can construct the first MPQC-PVIA protocol from Trap-code form AQA in the preprocessing model. The protocol will be very similar to the Clifford-form protocol (Protocol 4) but with Trap-form IE and AQA (Figure 4). Similar to [ACC+21], we make use of a verifiable quantum fully homomorphic encryption scheme (vQFHE) [ADSS17] for computation on Trap code.

Protocol 7 ($\Sigma^{\mathsf{BoBW_{TP}}}, \Pi^{\mathsf{BoBW_{TP}}}$) for the Trap-form Preprocessing MPQC-PVIA

Preprocessing: $\Sigma^{\mathsf{BoBW_{TP}}}$

1. Run Σ^{IE}, distributing not only Trap-code ciphertexts on EPR pairs and ancilla states, but also gadgets required for vQFHE computation under secret key sk given to cMPC.
2. Run $\Sigma^{\mathsf{AQA}}(\mathsf{P}_j, \mathsf{P}_i)$ for all pairs $i, j \in [n]$.

Protocol: $\Pi^{\mathsf{BoBW_{TP}}}$

1. All parties run Π^{IE}. so that each P_i obtains several Trap-code ciphertexts $\{\sigma_{i,l}\}$ corresponding to QECC codewords of inputs, under the same permutation key π_i.
2. cMPC initializes the set $\texttt{Corr} = \{\}$.
   At any time if $|\texttt{Corr}| > \mathsf{t}$, cMPC publicly abort to all parties in $\texttt{Corr}$.
3. P_i computes via $(\{\tilde{\sigma}_{i,l}\}, \mathsf{log}_i) \leftarrow \mathsf{TrapTP.Eval}(C, \{\sigma_{i,l}\})$ and sends log_i to cMPC.
4. cMPC runs $\mathsf{CheckLog}(\mathsf{log}_i, \mathsf{sk})$.If the output is $\perp$, cMPC publicly abort P_i.
5. P_i sends $\tilde{\sigma}_{i,l}$ to the respective receiver P_j via Π^{AQA}. The receiver P_j gets σ_j'.
6. cMPC sends the decoding instruction $V_{\mathsf{Dec},i}$ to the respective party P_i.

Output: P_i outputs $\rho_i' \leftarrow V_{\mathsf{Dec},i}(\sigma_i')$.

The Trap-form protocol $(\Sigma^{\mathsf{BoBW_{TP}}}, \Pi^{\mathsf{BoBW_{TP}}})$ satisfies the statement of Theorem 6.3. We omit the proof because of its similarity to the Clifford-code form MPQC-PVIA while basing on the security guarantee of Trap-code form AQA.

Acknowledgement

We would like to thank Kai-Min Chung and Jiapeng Zhang for their helpful discussions. This research is partially supported by NSTC QC project under Grant no. NSTC 111-2119-M-001-004- and partially supported by NSF CAREER award 2141536.

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.

[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.

[ADSS17] Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 438–467. Springer, 2017.

[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.

[BDO14] Carsten Baum, Ivan Damgård, and Claudio Orlandi. Publicly auditable secure multi-party computation. In *International Conference on Security and Cryptography for Networks*, pages 175–196. Springer, 2014.

[BDSW96] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Annual Cryptology Conference*, pages 344–360. Springer, 2013.

[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.

[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.

[BW16] Anne Broadbent and Evelyn Wainewright. Efficient simulation for quantum message authentication. In *International Conference on Information Theoretic Security*, pages 72–91. Springer, 2016.

[BY22] Zvika Brakerski and Henry Yuen. Quantum garbled circuits. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 804–817, 2022.

[CGL99] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical review letters*, 83(3):648, 1999.

[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.

[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.

[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.

[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.