Secure Multi-party Quantum Computation

Claude Crépeau [*], Daniel Gottesman [†], Adam Smith [‡]

Abstract

Multi-party computing, also called *secure function evaluation*, has been extensively studied in classical cryptography. We consider the extension of this task to computation with quantum inputs and circuits. Our protocols are information-theoretically secure, i.e. no assumptions are made on the computational power of the adversary. For the slightly weaker task of *verifiable quantum secret sharing*, we give a protocol which tolerates any $t < n/4$ cheating parties (out of n). This is shown to be optimal. We use this new tool to show how to perform any multi-party quantum computation as long as the number of dishonest players is less than $n/6$.

[*] School of Computer Science, McGill University, Montréal (Québec), Canada. e-mail: `crepeau@cs.mcgill.ca`. Supported in part by Québec's FCAR and Canada's NSERC.

[†] UC Berkeley, EECS: Computer Science Division, Soda Hall 585, Berkeley, CA 94720, USA. e-mail: `gottesma@eecs.berkeley.edu`. Supported by the Clay Mathematics Institute.

[‡] M.I.T., Laboratory for Computer Science, 200 Technology Square, Cambridge MA 02139, USA. e-mail: `asmith@theory.lcs.mit.edu`. Supported in part by U.S. Army Research Office Grant DAAD19-00-1-0177. Some of this research was done while the author was visiting McGill University.

1 Introduction

Secure distributed protocols have been an important and fruitful area of research for modern cryptography. In this setting, there is a group of participants who wish to perform some joint task, despite the fact that some of the participants in the protocol may cheat in order to obtain additional information or corrupt the outcome.

We investigate a quantum version of an extensively studied classical problem, *secure multi-party computation* (or *secure function evaluation*), first introduced by [18]. For this paper, we consider an extension of this task to quantum computers. A multi-party quantum computing (MPQC) protocol allows n participants $P_1, P_2, \ldots, P_n$ to compute an n-input quantum circuit in such a way that each party P_i is responsible for providing one (or more) of the input states. The output of the circuit is broken into n components $\mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_n$, such that P_i receives the output $\mathcal{H}_i$. Note that the inputs to this protocol are arbitrary quantum states—the player providing an input need only have it in his possession, he does not need to know a classical description of it. Moreover, unlike in the classical case, we cannot assume without loss of generality that the result of the computation will be broadcast. Instead, each player in the protocol receives some part of the output.

Informally, we require two security conditions:
- *Soundness and Completeness:* no coalition of t or fewer cheaters should be able to affect the outcome of the protocol beyond what influence they have by choosing their inputs.
- *Privacy:* no coalition of t or fewer cheaters should learn anything beyond what they can deduce from their initial knowledge of their input and from their part of the output.

Verifiable Quantum Secret Sharing. In order to construct MPQC protocols, we consider a subtask which we call *verifiable quantum secret sharing.* In classical cryptography, a verifiable secret sharing scheme [13] is a two phase protocol with one player designated as the "dealer". After the first phase (*commitment*), the dealer shares a secret amongst the players. In the second phase (*recovery*), the players reconstruct the value publicly.

The natural quantum version of this allows a dealer to share a state ρ (possibly unknown to him but nonetheless in his possession). Because quantum information is not clone-able, we cannot require that the state be reconstructed publicly; instead, the recovery phase also has a designated player, the reconstructor R. We require that, despite any malicious actions by up to t players:
- *Soundness:* As long as R is honest and the dealer passes the commitment phase successfully, then there is a unique quantum state which can be recovered by R.
- *Completeness:* When D is honest, then he always passes the commitment phase. Moreover, when R is also honest, then the value recovered by R is exactly D's input ρ.
- *Privacy:* When D is honest, no other player learns info about D's input until the recovery step.

Note that for quantum data, the privacy condition definition is redundant: any information obtained about the shared state would imply some disturbance of that state, contradicting the completeness requirement.

Contributions. We give a protocol for verifiable quantum secret sharing that tolerates any number $t < n/4$ of cheaters. We show that this is optimal, by proving that VQSS is impossible when $t \geq n/4$. Based on techniques from fault-tolerant quantum computing, we use our VQSS protocol to construct a multi-party quantum computation protocol tolerating any $t < n/6$ cheaters. (MPQC is similar to standard fault-tolerance but with a different error model, see Previous Work). Our protocols run in time polynomial in both n, the number of players, and k, the security parameter. The error of the protocols is exponentially small in k.

Beyond these specific results, there are a number of conceptual contributions of this paper to the theory of quantum cryptographic protocols. We provide a simple, general framework for defining and proving the security of distributed quantum protocols in terms of equivalence to an ideal protocol involving a third party. This follows the definitions for classical multi-party protocols, which have been the subject of considerable recent work [19, 5, 23, 7, 16, 9, 25, 8, 28]. The analysis of our protocols leads us to consider various notions of local "neighborhoods" of quantum states, and more generally of quantum codes. We discuss three notions of a neighborhood. The notion most often used for the analysis of quantum error-correction and fault-tolerance is insufficient for our needs, but we show that a very natural generalization (specific to so-called "CSS" codes) is adequate for our purposes. Along the way, we provide modified versions of the classical sharing protocols of [12]. The new property our protocols have is that dealers do not need to remember the randomness they use when constructing shares to distribute to other players. This allows them to replace a random choice of coins with the *superposition* over all such choices.

1.1 Previous Work

Classical MPC. Multi-party computing was introduced by Goldreich, Micali and Wigderson [18], who showed that *under computational assumptions*, secure multi-party evaluation of any function was possible tolerating any minority of cheating players, i.e. if and only if $t < \frac{n}{2}$. If one assumes pairwise secure channels but no computational assumptions, then one can compute any function securely if and only if $t < n/3$ [6, 12]). If one further assumes the availability of a secure broadcast channel, then one can in fact tolerate $t < n/2$, and no more ([26, 4, 15]). All of these protocols rely on verifiable secret sharing as a basic tool. Our solution draws most heavily on the techniques of Chaum, Crépeau and Damgård [12].

Beyond these basic protocols, much work has focused on finding proper definitions of security [19, 5, 23, 7, 16, 9, 25, 8]. We adopt a simple definition based on the initial definitions of Canetti.

Quantum Secret Sharing. Relatively little work exists on multi-party cryptographic protocols with quantum data. Secret sharing with a quantum secret was first studied by Cleve et al. [14], who showed an equivalence with quantum error-correcting codes (QECC). Their scheme is the basis of our protocols. Chau [11] deals with classical computations, but also mentions the problem of verifiable quantum secret sharing as an open question.

Fault-tolerant Quantum Computing. The goal of FTQC is to tolerate *non-malicious* faults occurring within a single computer. One assumes that at every stage in the computation, every qubit in the circuit has some known probability p of suffering a random error, i.e. of becoming completely scrambled. Moreover, errors are assumed to occur *independently* of each other and of the data in the computation.

One can view multi-party computation as fault-tolerant computing with a different error model, one that is suited to distributed computing. The MPQC model is weaker in some respects since we assume that errors will always occur in the same, limited number of positions, i.e. errors will only occur in the systems of the t corrupted players. In other respects, the error model of MPQC is stronger: in our setting errors may be *maliciously* coordinated. In particular, they will not be independently placed, and they may in fact depend on the data of the computation—the adversaries will use any partial information known about the other players' data, as well as information about their own data, to attempt to corrupt the computation. For example, several FTQC algorithms rely on the fact that at certain points in the computation, at most one error is likely to occur. Such algorithms will fail when errors are placed adversarially. Techniques from FTQC are nonetheless useful for multi-party computing. We will draw most heavily on techniques due to Aharonov and Ben-Or [3].

1.2 Definitions and Model

In this paper, we use a simple simulation-based framework for proving the security of quantum framework, similar to early classical definitions [7]. We specify a task by giving a protocol for implementing it in an ideal model where players have access to a trusted third party. We prove a given protocol secure by showing a simulator which translates any attack in the real-world protocol into an (almost) equally successful attack in the ideal model.

We assume that every pair of participants is connected by perfect (i.e. authenticated, secret) quantum and classical channels, and that there is a classical authenticated broadcast channel to which all players have access. Because we will always consider settings where $t < \frac{n}{2}$, we can also assume that players can perform *classical* multi-party computations securely [15][1]. The adversary is an arbitrary quantum algorithm (or family of circuits) $\mathcal{A}$ (not necessarily polynomial time), and so the security of our protocols does not rely on computational assumptions.

The real and ideal models, as well as the notion of security, are specified more carefully in Appendix A. For now, use the following informal definitions for the ideal protocols. The real protocols are secure if they succeed in simulating the ideal ones.

Multi-party Quantum Computation. All players hand their inputs to the trusted party, who runs the desired circuit and hands back the outputs. Note that the only kind of cheating which is possible is that cheaters may choose their own input. In particular, cheaters cannot force the protocol to abort.

Verifiable Quantum Secret Sharing. In the sharing phase, the dealer gives his secret system to the trusted party. In the reconstruction phase, the $\mathcal{TTP}$ sends the secret system to the reconstructor R. The only catch is that in the ideal model, honest players should not learn the identity of R until after the first phase has finished (otherwise, D could simply send the secret state to R in the first phase without violating the definition).

1.3 Preliminaries

We present the notation necessary for reading the protocols and proofs in this paper. For a more detailed explanation of the relevant background, see Appendix B, or a textbook such as [24].

We will work with p-dimensional quantum systems, for some prime $p > n$. Such a system is called a qupit, and the "computational" basis states are labelled by elements in $\mathbb{Z}_p$. We will also be working in the Fourier basis, which is given by the unitary transformation $\mathcal{F}|a\rangle \mapsto \sum_b \omega^{ab}|b\rangle$ (ignoring normalizing constants). A basis for the operators on a qupit is given by the p^2 Pauli operators X^aZ^b, where $X|a\rangle = |a+1\rangle$, $Z|a\rangle = \omega^a|a\rangle$, and $\omega = exp(2\pi i/p)$. Tensor products of these operators yield the Pauli basis for the set of operators on a register of qupits. The weight of a tensor product operator is the number of components in which it is not the identity $\mathbb{I}$.

Quantum Codes. The quantum error-correcting codes used in this paper are CSS codes. These are defined via two classical linear codes $V, W \subseteq \mathbb{Z}_p^n$ such that $V^\perp \subseteq W$. If we denote $W^{(q)} = \text{span}\{|\mathbf{w}\rangle : \mathbf{w} \in W\}$ for a classical code W, then we can write the CSS code as $\mathcal{C} = V^{(q)} \cap \mathcal{F}W^{(q)}$. Thus, $\mathcal{C}$ is the set of states of n qubits which yield a codeword of V when measured in the computational basis and a codeword of W when measured in the Fourier basis.

Specifically, we will use quantum Reed-Solomon codes. We specify a quantum RS code by a single parameter $\delta < (n-1)/2$. The classical Reed-Solomon code V^δ is given by the vectors $\hat{\mathbf{q}} = (q(1), q(2), \ldots, q(n))$ for all univariate polynomials q of degree at most δ. The related code V_0^δ

[1] In fact, even the assumption of a broadcast channel is not strictly necessary, since $t < \frac{n}{3}$ in our setting.

is the subset of V^δ corresponding to polynomials which interpolate to 0 at the point 0. That is: $V^\delta = \{\hat{\mathbf{q}} : \ q \in F[x] : \ \deg(q) \leq \delta\}$ and $V_0^\delta = \{\hat{\mathbf{q}} : \ \deg(q) \leq \delta \text{ and } q(0) = 0\} \subseteq V^\delta$. The code V^δ has minimum distance $d = n - \delta$, and an efficient error-correction procedure. Let $\delta' = n - \delta - 1$. There are constants $d_1, ..., d_n \in \mathbb{Z}_p$ such that the dual of the code V^δ is just the code $V_0^{\delta'}$, rescaled by d_i in the i^{th} coordinate; similarly, the dual of V_0^δ is a rescaled version of $V^{\delta'}$. Denote these duals by $W_0^{\delta'}, W^{\delta'}$, respectively.

The quantum code $\mathcal{C}^\delta$ for parameter δ is the CSS code obtained from codes $V = V^\delta$ and $W = W^{\delta'}$. It encodes a single qupit, and has minimum distance $\delta + 1$ (thus, it corrects $t = \lfloor \delta/2 \rfloor$ errors). Moreover, errors can be corrected efficiently, given the syndrome of a corrupted codeword, i.e. the V-syndrome measured in the computational basis and the W-syndrome measured in the Fourier basis.

Transversal Operations. Certain logical operations on data encoded via CSS codes can be performed *transversally* (i.e. using only local operations, which affect the same component of two codewords). For example, applying a linear gate $|a\rangle|b\rangle \mapsto |a\rangle|a + cb\rangle$ (for a constant c) to each component of a pair of codewords applies that gate to the encoded data. Applying the Fourier transform transversally to a codeword yields a somewhat different effect: data encoded with the codes V, W is mapped to the Fourier transform of that data, encoded with the dual code $\tilde{\mathcal{C}}$ defined via the codes W, V. For quantum RS codes, rescaling each component of the dual code of $\mathcal{C}^\delta$ produces the code $\mathcal{C}^{\delta'}$, enabling us to perform the map $\mathcal{E}_{\mathcal{C}^\delta}|\psi\rangle \mapsto \mathcal{E}_{\mathcal{C}^{\delta'}}(\mathcal{F}|\psi\rangle)$, where $\mathcal{E}_\mathcal{C}$ is the encoding map for a code $\mathcal{C}$.

A nice result of fault-tolerant computing [3, 20] is that when $\delta < n/3$ (i.e. $t < n/6$), one can in fact perform any operation on data encoded by a quantum RS code using only local operations and classical information transmitted between the components. So long as the classical communication is reliable, then these procedures can tolerate arbitrary corruption of $n/6$ of the positions. These procedures are described in Appendix B.3.

2 Neighborhoods of Quantum Codes

One of the ideas behind classical multi-party computing protocols is to ensure that data is encoded in a state that remains "close" to a codeword, differing only on those positions held by cheaters (call that set B). For classical codes, "close" means that the real word $\mathbf{v}$ should differ from a codeword only on B, so that any errors introduced by cheaters are correctable. For a code W, let the B-neighborhood W_B be the set of vectors differing from a codeword of W by positions in B, i.e., $W_B = \{\mathbf{v} \ : \ \exists \mathbf{w} \in W \text{ s.t. } \mathrm{supp}(\mathbf{v} - \mathbf{w}) \in B\}$. Equivalently, one can define W_B as the set of words obtained by distributing a (correct) codeword to all players, and then having all players send their shares to some (honest) reconstructor R.

For quantum codes, there is more than one natural definition of the neighborhood corresponding to a set B of positions. Let $\{1, ..., n\}$ be partitioned according to two sets A, B. We say a mixed state ρ' is "in" $\mathcal{C}$ if all states in the mixture lie in $\mathcal{C}$, i.e. $\mathrm{Tr}(P_\mathcal{C}\rho') = 1$ where $P_\mathcal{C}$ is the projector onto $\mathcal{C}$. We consider three definitions of a "B-neighborhood" of a CSS code $\mathcal{C}$. Let ρ be an arbitrary state of the coding space.

1. ρ differs from a state in $\mathcal{C}$ only by some super-operator local to B:
   $N_B(\mathcal{C}) = \{\rho : \exists \rho' \text{ in } \mathcal{C}, \exists \mathcal{O} \text{ super-operator, acting only on } B \text{ s.t. } \rho = \mathcal{O}(\rho')\}$.

2. ρ cannot be distinguished from a state in $\mathcal{C}$ by looking only at positions in A.
   $ST_B(\mathcal{C}) = \{\rho : \exists \rho' \text{ in } \mathcal{C} \text{ s.t. } \mathrm{Tr}_B(\rho) = \mathrm{Tr}_B(\rho')\}$.

3. Specifically for CSS codes, one can require that the state ρ pass checks on A in both bases, i.e. that measuring either the V_B-syndrome in the computational basis, or the W_B-syndrome in the Fourier basis, yields 0. The set of states which pass this test is: $\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}$.

In general, these notions form a strict hierarchy: $N_B(\mathcal{C}) \subsetneq ST_B(\mathcal{C}) \subsetneq \mathcal{C}_B$. Only one of them—notion (3)—is always a subspace (see Appendix C for details).

In the analysis of quantum error-correction and fault-tolerance schemes, it is sufficient to consider notion (1), for two reasons. On one hand, one starts from a correctly encoded state. On the other hand, the errors introduced by the environment will be independent of the encoded data (and in fact they must be for error-correction to be possible at all in that context).

In our setting, however, we cannot make such assumptions, since the cheaters might possess states which are entangled with the data in the computation, and so the errors they introduce will not be independent of that data. Instead, we show that our verifiable sharing protocol guarantees a condition similar to notion (3) (see Lemma 3.1). In order to provide some intuition, we characterize notion (3) below. Proofs can be found in Appendix C.1.

Well-Definedness of Decoding for $\mathcal{C}_B$. The set $\mathcal{C}_B$ is a subspace, since it is defined in terms of measurement outcomes. More particularly, it is the space spanned by the states of $N_B(\mathcal{C})$:

Lemma 2.1. *If ρ is in $\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}$, then we can write $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where $|\psi_i\rangle = \sum_j c_{ij} E_j |\phi_{ij}\rangle$, the E_j are Pauli operators on B and $|\phi_{ij}\rangle \in \mathcal{C}$.*

This has a useful corollary, namely that decoding is well-defined for states in $\mathcal{C}_B$. Formally, there are two natural "reconstruction operators" for extracting the secret out of a state which has been shared among several players. Suppose that $\mathcal{C}$ has distance $d > 2t + 1$ and $|B| \leq t$. First, $\mathcal{D}$ is the decoding operator for the error-correcting code $\mathcal{C}$. For any operator E_j of weight less than t and for any state $|\bar{\phi}\rangle$ in $\mathcal{C}$, we have $\mathcal{D} E_j |\phi\rangle = |\phi\rangle \otimes |j\rangle$ (i.e. the error is not only corrected but also identified). It will then discard the system containing the syndrome information $|j\rangle$. Second, $\mathcal{R}^I$ is the "ideal recovery operator", defined by identifying the set B of cheaters and applying the simple interpolation circuit to any set of $n - 2t$ good players' positions.

Proposition 2.2. *For any state ρ in $\mathcal{C}_B$, the state $\mathcal{R}^I(\rho)$ is well-defined and is equal to $\mathcal{D}(\rho)$.*

Our protocols guarantee conditions similar to $\mathcal{C}_B$, and the well-definedness is essential for proving simulatability.

3 VQSS Protocol: Two-Level Quantum Sharing

In this section we define a two-tiered protocol for VQSS. It is based on the VQSS protocols of [12] as well as on the literature on quantum fault-tolerance and error-correction, most notably on [3]. Detailed proofs for the claims of this section are in Appendix D.

3.1 Sharing Shares: 2-GOOD Trees

In the VSS protocol of [12], the dealer D takes his secret, splits it into n shares and gives the i^{th} component to player i. Player i then shares this secret by splitting it into n shares and giving the j^{th} share to player j. Thus, there are n^2 total shares, which can be thought of as the leaves of a tree with depth 2 and fan-out n: each leaf is a share; the i^{th} branch corresponds to the shares created by player i, and the root corresponds to the initial shares created by the dealer. Player j

holds the j^{th} leaf in each branch of this tree. We will run a cut-and-choose protocol in order to guarantee some kind of consistency of the distributed shares.

During the protocol we accumulate $n+1$ sets of apparent cheaters: one set B for the dealer (this corresponds to a set of branches emanating from the root), and one set B_i for each player i (this corresponds to a subset of the leaves in branch i). These sets all have size at most t. At the end of the protocol, we want to guarantee certain invariants. Say V has minimum distance $> 2t$, and each codeword corresponds to a single value $a \in \mathbb{Z}_p$.

Definition 1 (2-GOOD trees). *We say a tree of n^2 field elements is 2-GOOD with respect to the code V and the sets $B, B_1, ..., B_n$ if:*

1. *For each $i \notin C$ (i.e., corresponding to an honest player), we have $B_i \subseteq C$, i.e. all apparent cheaters are real cheaters.*
2. *For each branch $i \notin B$, the shares held by the honest players* not in *B_i should all be consistent with some codeword in V. That is, the vector of all shares should be in $V_{B_i \cup C}$, where C is the set of cheating players.*

   N.B.: Because there are at most t players in B_i and at most t cheaters, there are at least $d+1 \leq n-2t$ honest players remaining, and so the polynomial above is uniquely defined. This guarantees that for each branch $i \notin B$, there is a unique value $a_i \in F$ which is obtained by interpolating the shares of the honest players not in B_i.
3. *For $i \notin B$, the values a_i defined by the previous property are all consistent with a codeword of V (i.e. the vector $(a_1, ..., a_n)$ is in V_B).*

We will abbreviate this as 2-GOOD$_V$, when the sets $B, B_1, ..., B_n$ are clear from the context.

3.2 VQSS Protocol

The VQSS protocol is described in Protocols 3.1 and 3.2. Intuitively, it guarantees that a tree of quantum shares would yield a 2-GOOD tree of classical values if measured in either the computational basis or the Fourier basis. We use the codes $V = V^\delta = V^{\delta'}$ and $W = W^\delta = W^{\delta'}$, with $n = 4t+1, \delta = \delta' = 2t$, although there is in fact no need to do this: the protocol will work for any CSS code with distance at least $2t+1$, so long as the codes V, W are efficiently decodable.

The protocol can be tweaked for efficiency. The final protocol takes three rounds. Each player sends and receives $O(n + \log \frac{1}{\epsilon})$ qubits, and the broadcast channel is used $O\left(n(n + \log \frac{1}{\epsilon})\right)$ times overall, where ϵ is the soundness error of the protocol (this requires setting $k = n + \log(\frac{1}{\epsilon})$).

Why is this a secure VQSS protocol? We want to show that the protocol is equivalent to the "ideal model", where at sharing time the dealer sends his secret system S to a trusted outside party, and at reveal time the trusted party sends S to the designated receiver. To do that, we will use two main technical claims (one for soundness, one for completeness).

Soundness. We must show that at the end of the protocol, if the dealer passes all tests then there is an well-defined "shared state" which will be recovered by the dealer. To do so, we guarantee a property similar to $\mathcal{C}_C$ (notion (3) of Section 2).

Protocol 3.1 (VQSS—**Sharing Phase).** Dealer D gets as input a quantum system S to share.

- **Sharing:**

  1. The dealer D prepares $(k+1)^2$ systems of n qupits each, called $S_{\ell,m}$ (for $\ell = 0, ..., k$ and $m = 0, ..., k$):
     (a) Encodes S using $\mathcal{C}$ in $S_{0,0}$.
     (b) Prepares k systems $S_{0,1}, ..., S_{0,k}$ in the state $\sum_{a \in F} \mathcal{E}_{\mathcal{C}} |a\rangle = \sum_{v \in V} |v\rangle$.
     (c) Prepares $k(k+1)$ systems $S_{\ell,m}$, for $\ell = 1, ..., k$ and $m = 0, ..., k$, each in the state $|\bar{0}\rangle = \sum_{v \in V_0} |v\rangle$.
     (d) For each of the $(k+1)^2$ systems $S_{\ell,m}$, D sends the i^{th} component (denoted $S^{(i)}_{\ell,m}$) to player i.
  2. Each player i, for each $\ell, m = 0, ...k$:
     (a) Encodes the received system $S^{(i)}_{\ell,m}$ using $\mathcal{C}$ into an n qupit system $S_{\ell,m,i}$.
     (b) Sends the j^{th} component $S^{(j)}_{\ell,m,i}$ to player j.

- **Verification:**

  1. Get public random values $b_1, ..., b_k \in_R F$. For each $\ell = 0, ..., k$, $m = 1, ..., k$, each player j:
     (a) Applies the controlled-addition gate (c-X^{b_m}) to his shares of the systems $S_{\ell,0,i}$ and $S_{\ell,m,i}$.
     (b) Measures his share of $S_{\ell,m,i}$ and broadcasts the result
     (i.e. each player broadcasts $k(k+1)n$ values).
     (c) For each $i \in \{1, ..., n\}$, players update the set B_i based on the broadcast values: there are $(k+1)kn$ broadcast words $\mathbf{w}_{\ell,m,i}$. Applying classical decoding to each of these yields min-weight error vectors $\mathbf{e}_{\ell,m,i}$ with support set $B_{\ell,m,i}$. Set $B_i = \cup_{\ell,m} B_{\ell,m,i}$. If there are too many errors, instead add i to the global set B.
     (d) Furthermore, players do the same at the root level: for all $i \notin B$, there is an interpolated value a_i which corresponds to the decoded codeword from the previous step. Players also decode the codeword $(a_1, ..., a_n)$ and update B accordingly (i.e. by adding any positions where errors occur to B).
  2. All players apply the Fourier transform $\mathcal{F}$ to their shares.
  3. Get public random values $b'_1, ..., b'_k \in_R F$. For $\ell = 1, ..., k$, each player j:
     (a) Applies the controlled-addition gate (c-$X^{b'_\ell}$) to his shares of the systems $S_{0,0,i}$ and $S_{\ell,0,i}$.
     (b) Measures his share of $S_{\ell,0,i}$ and broadcasts the result
     (i.e. each player broadcasts kn values).
     (c) For each $i \in \{1, ..., n\}$, players update B_i and B based on the broadcast values (as in Step 1c).
     [Note: the sets B and $B_1, ..., B_n$ are cumulative throughout the protocol.]
  4. All players apply the inverse transform $\mathcal{F}^{-1}$ to their shares of $S_{0,0}$.

- The remaining shares (i.e. the components of the n systems $S_{0,0,i}$) form the sharing of the state ρ.

Protocol 3.2 (VQSS—**Reconstruction Phase).** Player j sends his share of each of the systems $S_{0,0,i}$ to the receiver R, who runs the following decoding algorithm:

1. For each branch i: Determine if there is a set $\tilde{B}_i$ such that $B_i \subseteq \tilde{B}_i$, $\left|\tilde{B}_i\right| \leq t$ and the shares of $S_{0,0,i}$ lie in $\mathcal{C}_{\tilde{B}_i}$.
   If *not*, add i to B.
   Otherwise, correct errors on $\tilde{B}_i$ and decode to obtain a system S'_i.
2. Apply interpolation to any set of $n - 2t$ points not in B. Output the result S'.

Lemma 3.1. *The system has high fidelity to the following statement: "Either the dealer is caught ($|B| > t$) or measuring all shares in the computational (resp. Fourier) basis would yield a 2-GOOD tree with respect to the code V (resp. W)."*

Proof of this is via a "quantum-to-classical" reduction, similar to that of [22]. First, checks in the computational and Fourier bases don't interfere with each other, since they commute for CSS codes. Second, in a given basis, we can assume w.l.o.g. that all ancillae are first measured in that basis, reducing to a classical analysis similar to [12].

Ideal Reconstruction. In order to prove soundness carefully, we define an *ideal interpolation* circuit $\mathcal{R}^I$ for 2-GOOD trees: pick the first $n-2t$ honest players not in B, say $i_1, ..., i_{n-2t}$. For each i_j, pick $n-2t$ honest players not in B_{i_j} and apply the normal interpolation circuit (i.e. erasure-recovery circuit) for the code to their shares to get some qupit R_{i_j}. Applying the interpolation circuit again, we extract some system S which we take to be the output of the ideal interpolation. The following lemma then applies, following essentially from Proposition 2.2.

Lemma 3.2. *Given a tree of qupits which is 2-GOOD in both bases, the output of the ideal interpolation and the real recovery operators are the same. In particular, this means that no changes made by cheaters to their shares of a 2-GOOD tree can affect the outcome of the recovery operation.*

Lemmas 3.1 and 3.2 together imply that there is essentially a unique state which will be recovered in the reconstruction phase when the receiver R is honest. Thus, Protocol 3.1 is sound.

Completeness. As discussed earlier, the protocol is considered complete if when the dealer is honest, the state that is recovered by an honest reconstructor is exactly the dealer's input state. The key property is that *when the dealer D is honest, the effect of the verification phase on the shares which never pass through cheaters' hands is the identity.*

Consider the case where the dealer's input is a pure state $|\psi\rangle$. On one hand, we can see by inspection that an honest dealer will always pass the protocol. Moreover, since the shares that only go through honest players' hands remain unchanged, it must be that if some state is reconstructed, then that state is indeed $|\psi\rangle$, since the ideal reconstruction operator uses only those shares. Finally, we know that since the dealer passed the protocol the overall tree must be 2-GOOD in both bases, and so some value will be reconstructed. Thus, on input a pure state $|\psi\rangle$, an honest reconstructor will reconstruct $|\psi\rangle$. We have proved:

Lemma 3.3. *If D and R are honest, and the dealer's input is a pure state $|\psi\rangle$, then R will reconstruct a state ρ with fidelity $1 - 2^{-\Omega(k)}$ to the state $|\psi\rangle$.*

Not surprisingly, this lemma also guarantees the privacy of the dealer's input. By a strong form of the no cloning theorem, any information the cheaters could obtain would cause some disturbance, at least for a subset of the inputs. Thus, the protocol is in fact also private.

Simulatability. The claims above show that the protocol satisfies an intuitive notion of security. We can prove security more formally by giving a simulator which converts any real-model adversary into an ideal-model one. The idea is that the simulator will simulate the regular VQSS protocol either on input provided by a cheating dealer or on bogus data $|0\rangle$, and then extract and/or change the shared state as needed.

Theorem 3.4. *Protocol 3.1 is a statistically secure implementation of* VQSS *(Protocol A.2).*

The simulation is given in Section D.3.3. The key is showing that any entanglement between data and errors can be produced by the cheaters "on their own", i.e. in the ideal model with a $\mathcal{TTP}$.

3.3 Additional Properties of Two-Level Sharing

Two-level sharings produced by the same dealer (using the protocol above) have some additional properties, which will be useful for multi-party computation. First of all, notice that there is no problem in tracking the sets $B, B_1, ..., B_n$ incrementally across various invocations of the protocol for the same dealer, and so we assume below that these sets are the same for different sharings from the same dealer.

1. Some operations can be applied transversally to a valid sharings. Applying the linear operation $(x, y) \mapsto (x, y + bx)$ (denoted $c\text{-}X^b$) to all shares of two sharings effectively applies $c\text{-}X^b$ to the shared states. Similarly, applying the Fourier rotation transversally changes the sharing to the dual code and applies a logical Fourier rotation. Finally, measuring all shares of a valid sharing in the computational basis and applying classical decoding yields the same result as measuring the shared state. Thus, players can measure without exchanging quantum information.
2. The dealer can use the protocol to additionally prove to all players that the system he is sharing is the exactly the state $|0\rangle$: the ancillas he uses in this case will all be sharings of $|0\rangle$ (instead of $\sum |a\rangle$). The verification step is the same as before, except now players verify that the reconstructed codeword at the top level interpolates to 0. Similarly, the dealer can prove that he is sharing a state $\sum_a |a\rangle$. This will be useful for sharing ancillas in the MPQC protocol.

4 Impossibility of VQSS when $t \geq \frac{n}{4}$

Lemma 4.1. *No* VQSS *scheme exists for 4 players which tolerates one cheater.*

Proof: Suppose such a scheme exists. Consider a run of the protocol in which all players behave perfectly honestly until the end of the sharing phase, at which one (unknown) player introduces an arbitrary error. However, an honest "receiver" Ruth, given access to the state of all players, must still be able to recover the shared state. Thus, the joint state of all players constitutes a four-component QECC correcting one error. However, no such code exists, not even a mixed-state one, by the quantum Singleton bound (Appendix B.2). □

The optimality of our VQSS scheme is an immediate corollary, since any protocol tolerating $n/4$ cheaters could be used to construct a four-person protocol tolerating one cheater by having each participant simulate $n/4$ players in the original protocol:

Theorem 4.2. *No* VQSS *scheme for n players exists which tolerates all coalitions of $\lceil n/4 \rceil$ cheaters.*

Note that we have only proved the impossibility of *perfect* VQSS protocols. However, both the no cloning theorem and the equivalence of t-error-correction and $2t$-erasure-correction hold when exact equality is replaced by approximate correctness, and so in fact even statistical VQSS schemes are impossible when $t \geq n/4$.

5 Secure Multi-party Quantum Computation

In this section we show how to use the VQSS protocol of the previous section to construct a multi-party quantum computing scheme. First, we give a modified VQSS protocol. At the end of the protocol, all players hold a single qupit. With high fidelity, either the dealer will be caught cheating or the shares of all honest players will be consistent in both the computational and Fourier bases, i.e. there is no set B of "apparent cheaters". We then apply fault-tolerant techniques to achieve secure distributed computation.

5.1 Level 3 Sharing Protocol

Until now, we have used protocols for tolerating $t < n/4$ cheaters. However, we are now interested in tolerating $t < n/6$ cheaters. Thus, we take $n = 6t + 1$ for simplicity, and as before we set $\delta = 2t$ (thus $\delta' = 4t$). We will work with the CSS code $\mathcal{C}$ given by $V = V^{\delta}$ and $W = W^{\delta'}$. Recall that this is the CSS code for which there exist nearly-transversal fault-tolerant procedures (Section 1.3). Our goal is to share a state so that at the end all shares of honest players lie in $\mathcal{C}_C = V_C^{(q)} \cap \mathcal{F}^{\otimes n} W_C^{(q)}$.

The new scheme is given in Protocol 5.1. The idea is that the previous VQSS scheme allows distributed computation of linear gates and Fourier transforms on states shared by the same dealer. It also allows verifying that a given shared state is either $|0\rangle$ or $\sum |a\rangle$. The players will use this to perform a distributed computation of the encoding gate for the code $\mathcal{C}$. Thus, the dealer will share the secret system S, as well as δ states $\sum |a\rangle$ and $n - \delta - 1$ states $|0\rangle$. Players then apply the (linear) encoding gate, and each player gets sent all shares of his component of the output. As before, the main lemmas are soundness and completeness of the protocol:

Lemma 5.1 (Soundness). *At the end of the sharing phase, the system has high fidelity to "either the dealer is caught or the players' shares $S_1, ..., S_n$ lie in $\mathcal{C}_C$".*

Lemma 5.2 (Completeness). *When D is honest, on pure state input $|\psi\rangle$, the shared state will have high fidelity to* $\mathrm{span}\,\{\mathcal{E}|\psi\rangle\}_C$ *(i.e. will differ from $\mathcal{E}|\psi\rangle$ only on the cheaters' shares).*

Note the dealer can also prove that he has shared a $|0\rangle$ state (by showing that his input is $|0\rangle$).

5.2 Distributed Computation

Given the protocol of the previous section, and known fault-tolerant techniques, there is a natural protocol for secure multi-party computation of a circuit: have all players distribute their inputs via the top-level sharing (Protocol 5.1); apply the gates of U one-by-one, using the (essentially) transversal implementation of the gates described in Section B.3; then have all players send their share of each output to the appropriate receiver. See Protocol 5.2.

The only sticking point in the analysis is that the fault-tolerant procedures require some interaction, namely measuring a shared state. All players measure their share and broadcast the result, applying classical decoding to the resulting word. If the errors occurring in the measured ancilla were somehow correlated or entangled with errors in the real data, one could imagine that measuring and broadcasting them might introduce further entanglement. However, this will not be a problem: on one hand, any errors will occur only in the cheaters shares, and so provide nothing beyond what the cheaters could learn themselves; on the other hand, the honest players will discard all the information from the broadcast except the decoded measurement result (each honest player performs the decoding locally based on the broadcast values, so all honest players obtain the same result). Again, the cheaters can do this themselves.

Lemma 5.3. *Suppose that all inputs and ancillas are shared at the beginning via states in $\mathcal{C}_C$. Then the result of applying the protocol for a given circuit U, and then sending all states to an honest decoder R is the same as sending all states to R and having R apply U to the reconstructed states.*

Theorem 5.4. *For any circuit U, Protocol 5.2 is a statistically secure real-world implementation of multi-party quantum computation (Protocol A.1) as long as $t < n/6$.*

Proof of this is by simulation, as before. The simulator runs the sharing protocol, extracts the shared states and sends them to the $\mathcal{TTP}$. The results returned by the $\mathcal{TTP}$ are subsequently "swapped" back in, and the resulting share sent to the appropriate cheater. See Section D.4.

6 Open Questions

Given our results, the most obvious question is whether or not MPQC is possible when $n/6 \leq t < n/4$. One approach to this problem is to find a fault-tolerant Toffoli procedure for the code $\mathcal{C}^\delta$ for $n = 2\delta + 1$, which tolerates t errors at *any* point in the computation (known procedures [27, 3] have one point at which at most one error can be tolerated). Another natural direction of research is to find a VQSS protocol with zero error. For example, the techniques of [6] for the classical case do not seem to apply to the quantum setting. Finally, one can ask what tasks are achievable when we allow cheating players to force the abortion of the protocol (usually called an "optimistic protocol"). In that setting VQSS becomes largely irrelevant since there is no guarantee that the honest players can reconstruct the secret without the cheaters help. Thus, the bound of $n/4$ no longer seems hard; in fact, we conjecture that some improvement is possible, possibly even up to tolerating any minority of cheating players.

Acknowledgements

Thanks to Richard Cleve for helpful discussions. A.S. also thanks Madhu Sudan for patience, support and advice.

References

[1] *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, Chicago, Illinois, 2–4 May 1988.

[2] D. Aharonov and M. Ben-Or. Fault tolerant quantum computation with constant error. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 176–188, El Paso, Texas, 4–6 May 1997. This is a preliminary version of [3].

[3] D. Aharonov and M. Ben-Or. Fault tolerant quantum computation with constant error rate. Los Alamos eprint quant-ph/9906129. Journal version of [2] (submitted to SIAM J. Comp.), June 1999.

[4] D. Beaver. Multiparty protocols tolerating half faulty processors. In G. Brassard, editor, *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 560–572. IACR, Springer-Verlag, 1990, 20–24 Aug. 1989.

[5] D. Beaver. Foundations of secure interactive computing. In Feigenbaum [17], pages 377–391.

[6] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In ACM [1], pages 1–10.

[7] R. Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[8] R. Canetti. A unified framework for analyzing security of protocols. Manuscript, preliminary version available at eprint.iacr.org/2000/067. Some versions of the manuscript are titled "Universal Composability.", 2001.

Protocol 5.1 (Top-Level Sharing). Dealer D takes as input a qupit S to share.

- 1. **(Distribution)** The dealer D:
    - (a) Runs the level 2 VQSS protocol on input S.
    - (b) For $i = 1, ..., \delta$:
Runs level 2 sharing protocol to share state $\sum_a |a\rangle$ (see Remark 2 in Section 3.3)
    - (c) For $i = 1, ..., n - \delta - 1$:
Runs level 2 sharing protocol to share state $|0\rangle$ (see Remark 2 in Section 3.3)

   Denote the n shared systems by $S_1, ..., S_n$ (i.e. S_1 corresponds to S, $S_2, ..., S_{\delta+1}$ correspond to $\sum_a |a\rangle$ and $S_{\delta+2}, ..., S_n$ correspond to $|0\rangle$). Note that each S_i is a two-level tree, and thus corresponds to n components in the hands of each player.

   2. **(Computation)** Collectively, the players apply the Vandermonde matrix to their shares of $S_1, ..., S_n$. (If D is honest then system S_i encodes the i-th component of an encoding of the input S).
   3. For each i, all players send their shares of S_i to player i.
- **Quantum Reconstruction** Input to each player i is the share S_i and the identity of the receiver R.
   1. Each player i sends his share S_i to R.
   2. R outputs $\mathcal{D}(S_1, ..., S_n)$ and discards any ancillas.

Protocol 5.2 (Multi-party Quantum Computation).

1. **Input Phase:**
    - (a) For each i, player i runs Top-Level Sharing with input S_i.
    - (b) If i is caught cheating, then some player who has not been caught cheating yet runs Top-Level Sharing (Protocol 5.1), except this time with the one-dimensional code span$\{\mathcal{E}_{\mathcal{C}}|0\rangle\}$ (i.e. he proves that the state he is sharing is $|0\rangle$). If the sharing protocol fails, then another player who has not been caught cheating runs the protocol. There will be at most t iterations since an honest player will always succeed.
    - (c) For each ancilla state $|0\rangle$ needed for the circuit, some player who has not been caught cheating yet runs Top-Level Sharing (Protocol 5.1), with the one-dimensional code span$\{\mathcal{E}_{\mathcal{C}^{\delta}}|0\rangle\}$ or span$\{\mathcal{E}_{\mathcal{C}^{\delta'}}|0\rangle\}$, as needed. If the protocol fails, another player performs the sharing, and so forth.
2. **Computation Phase:** For each gate g in the circuit, players apply the appropriate fault-tolerant circuit, as described in Section B.3. Only the measurement used in Degree Reduction is not transversal. To measure the ancilla:
    - (a) Each player measures his component and broadcasts the result in the computational basis.
    - (b) Let $\mathbf{w}$ be the received word. Players decode $\mathbf{w}$ (based on the scaled Reed-Solomon code $W^{\delta'}$), and obtain the measurement result b.
3. **Output Phase:** For the i^{th} output wire:
    - (a) All players send their share of the output wire to player i.
    - (b) Player i applies the decoding operator for $\mathcal{C}$ and outputs the result. If decoding fails (this will occur only with exponentially small probability), player i outputs $|0\rangle$.

[9] R. Canetti, I. Damgrd, S. Dziembowski, Y. Ishai, and T. Malkin. On adaptive vs. non-adaptive security of multiparty protocols. In *Advances in Cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 262–279. IACR, Springer-Verlag, 2001.

[10] R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *Proceedings of the Twenty-Eigth Annual ACM Symposium on the Theory of Computing*, pages 639–648, 1996.

[11] H. F. Chau. Quantum-classical complexity-security tradeoff in secure multiparty computations. *Physical Review A*, 61, March 2000.

[12] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In ACM [1], pages 11–19.

[13] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 383–395, Portland, Oregon, 21–23 Oct. 1985. IEEE.

[14] R. Cleve, D. Gottesman, and H.-K. Lo. How to share a quantum secret. *Physical Review Letters*, 83:648–651, 1999.

[15] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations with dishonest minority. In J. Stern, editor, *Advances in Cryptology—EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*. IACR, Springer-Verlag, 1999.

[16] Y. Dodis and S. Micali. Parallel reducibility for information-theoretically secure computation. In M. Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 74–92. IACR, Springer, 2000.

[17] J. Feigenbaum, editor. *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. IACR, Springer-Verlag, 1992, 11–15 Aug. 1991.

[18] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 218–229, New York City, 25–27 May 1987.

[19] S. Goldwasser and L. A. Levin. Fair computation of general functions in presence of immoral majority. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology—CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 77–93. IACR, Springer-Verlag, 1991, 11–15 Aug. 1990.

[20] D. Gottesman. Personal communication. 2001.

[21] D. Gottesman and I. Chuang. Quantum teleportation is a universal computational primitive. *Nature*, November 1999.

[22] H.-K. Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283(5410):2050–2056, 26 Mar. 1999.

[23] S. Micali and P. Rogaway. Secure computation (abstract). In Feigenbaum [17], pages 392–404.

[24] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[25] B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In *ACM Conference on Computer and Communications Security 2000*, pages 245–254, 2000.

[26] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 73–85, Seattle, Washington, 15–17 May 1989.

[27] P. W. Shor. Fault-tolerant quantum computation. In *37th Annual Symposium on Foundations of Computer Science*, pages 56–65, Burlington, Vermont, 14–16 Oct. 1996. IEEE.

[28] J. van de Graaf. *Towards a formal defintion of security for quantum protocols.* PhD thesis, Université de Montréal, Canada, Dec. 1997. Available from http://www.cenapad.ufmg.br/˜jvdg/.

A Definitions

This section describes a simple framework for proving the security of distributed quantum cryptographic protocols. The definitions are based on the initial framework of Canetti [7], as well as on discussions in the dissertation of van de Graaf [28]. We describe two models for protocols. The first one–the "real" model—describes the environment we ultimately expect our protocols to run in. The second model is an idealized model in which players can interact with an incorruptable outside party. We will prove our "real-model" protocols secure by showing that they are equivalent to a simple protocol for the ideal model which captures our notion of what security means for a given task.

We provide no general composition theorems in this work. Instead, we simply prove the security of our composed protocols directly.

A.1 "Real" Model for Protocols

For the protocols in this paper, we assume that every pair of players is connected by perfect (i.e. authenticated, secret) quantum and classical channels. Moreover, we assume that there is a classical authenticated broadcast channel to which all players have access. Because we will consider settings where $t < \frac{n}{4} < \frac{n}{3}$, we can also assume that players can perform *classical* multi-party computations [6, 12][2].

The adversary is an arbitrary quantum algorithm (or family of circuits) $\mathcal{A}$. We make no assumptions about the computational power of the adversary; he is limited only by the number of players t that he can corrupt.

The *initial configuration* for the protocol is the joint state ρ of $n+2$ quantum systems: an input system $\mathcal{I}_i$ for each player in the protocol ($i = 1, ..., n$), as well as the adversary's auxiliary input system $\mathcal{I}_{aux}$ and an outside reference system $\mathcal{I}_{ref}$ (which will remain untouched throughout the protocol). Note that the input can be an arbitrary quantum state, possibly entangling all these systems.

A run of a "real model" protocol begins with all players receiving their input system $\mathcal{I}_i$ and the adversary receiving the state $\mathcal{I}_{aux}$. The adversary then chooses a subset C of size at most t of players to corrupt. From then on, the adversary has access to the state of the players in C and controls what they send over the channels. The adversary may cause the cheaters' systems to interact arbitrarily. His only restriction is that he has no access to the state of the honest players,

[2] In fact, even the assumption of a broadcast channel is unnecessary but (since $t < \frac{n}{3}$) but is made for simplicity.

and cannot intercept their communication. The reference system $\mathcal{I}_{ref}$ is untouched during this process.

At the end of the protocol, all players produce an output (for honest players, this is the output specified by the protocol). The system output by player i is denoted $\mathcal{O}_i$. Moreover, the adversary outputs an additional system $\mathcal{O}_{aux}$. The *output configuration* for the run of the protocol is the joint state of $\mathcal{O}_1, ..., \mathcal{O}_n$, the adversary's state $\mathcal{O}_{aux}$ and the reference system $\mathcal{I}_{ref}$. This state depends on the adversary $\mathcal{A}$ and the initial configuration ρ, and is denoted $Real(\mathcal{A}, \rho)$. Note that this configuration does not include any ancillary states or workspace used by honest players, only the output specified by the protocol (i.e. all other parts of the honest players' systems are "traced out").

A.2 "Ideal" Model For Protocols

The main difference of the ideal model from the real model is that there is a trusted third party (denoted $\mathcal{TTP}$) who helps the players in the execution of some protocol. The communications model is the same as before, except that every player is connected to $\mathcal{TTP}$ via a perfect (i.e. authentic, secret) quantum channel. There is no need to assume a broadcast channel since players can simply give a classical value to $\mathcal{TTP}$ and ask that it be re-sent to all players.

As before, the initial configuration consists of n systems $\mathcal{I}_i$ containing the players' inputs as well as the two systems $\mathcal{I}_{aux}$ and $\mathcal{I}_{ref}$. The $\mathcal{TTP}$ gets no input. The protocol proceeds as in the real model, except that players may interact with the $\mathcal{TTP}$, who may not be corrupted by the adversary. Finally, the output configuration is the same as before. The final state of the $\mathcal{TTP}$ is not included in the output configuration. The output configuration for adversary $\mathcal{A}$ and initial configuration ρ is denoted $Ideal(\mathcal{A}, \rho)$.

A.3 Protocol Equivalence

Suppose we have a protocol π which is supposed to implement some ideal functionality f, that is f is an ideal model protocol and π is an attempt to implement it in the real model.

Informally, we say π implements f if the input/output behavior of π cannot be distinguished from that of f. Formally:

Definition 2 (Perfect security). *A protocol π is considered perfectly secure if for all adversaries $\mathcal{A}_1$, there exists an adversary $\mathcal{A}_2$, running in time polynomial in that of $\mathcal{A}_1$, such that for all input configurations ρ (possibly mixed or entangled), we have:*

$$Real(\mathcal{A}_1, \rho) = Ideal(\mathcal{A}_2, \rho)$$

The protocols we design do not in fact achieve this strong notion of security. Instead, they take a security parameter k as input. All players receive the classical string 1^k as part of their input (in the ideal model, so does the $\mathcal{TTP}$). Moreover, the inputs may additionally depend on k (in particular, we allow the adversary's auxiliary input to depend on k). Since honest players should be polynomial-time quantum circuits, the protocol will run in time polynomial in k, although the adversary need not.

Definition 3 (Statistical security). *A protocol π is considered statistically secure if for all adversaries $\mathcal{A}_1$, there exists an adversary $\mathcal{A}_2$, running in time polynomial in that of $\mathcal{A}_1$, such that for all* sequences *of input configurations $\{\rho_k\}$ (possibly mixed or entangled), we have:*

$$F\left(Real(1^k, \mathcal{A}_1, \rho_k),\ Ideal(1^k, \mathcal{A}_2, \rho_k)\right) \geq 1 - 2^{-k},$$

where F denotes the fidelity of two quantum density matrices.

Simulators. Our definition asks us to construct a new adversary $\mathcal{A}_2$ for every real adversary A_1. To do so, we will follow the standard cryptographic paradigm of constructing a *simulator* $\mathcal{S}$ who uses $\mathcal{A}_1$ as a black box. Thus we can write $\mathcal{A}_2 = \mathcal{S}^{\mathcal{A}_1}$. We can view $\mathcal{S}$ as an "interface" between the real-world adversary and the ideal-model protocol [28]: $\mathcal{S}$ exchanges messages with $\mathcal{A}_1$, but must also control the corrupted parties in the ideal-model protocol.

When $\mathcal{A}_2$ is constructed in this way, then the definition above can be restated: Suppose that at the end of the protocol the adversary gains access to the outputs of the honest players. There should not exist a real-world adversary $\mathcal{A}_1$ that can tell the difference between (a) a run of the real protocol and (b) a run of the ideal-model protocol with $\mathcal{S}$ as an interface. We will construct simulators for our protocols in Section D.3.3 and Section 5.2.

A.4 Static versus Adaptive Adversaries

In this paper, we consider only static adversaries, who choose the parties they will corrupt before the beginning of the protocol and remain with that choice. On the other hand, an *adaptive* adversary chooses which players to corrupt as the protocol is progressing. The set of corrupted parties is still monotone—we do not allow a player to become honest again once he has been corrupted[3]—but the adversary can base his decision on the message he is seeing in the protocol. For example, if the players were to elect a small group of participants to make some decision amongst themselves, an adaptive adversary could wait until the selection had been made and then corrupt the members of that small group. Proving protocols secure against adaptive adversaries has been problematic even in the classical setting [10, 15].

Choosing to handle only static adversaries simplifies the definitions and proofs considerably, and offers no real loss of intuition. Nonetheless, we believe that the protocols we describe here are secure against adaptive adversaries, assuming that the environment in which the protocol is running somehow records which parties were corrupted and in what order (it is unclear what adaptivity even means without such an assumption). In Section D.1, we discuss briefly how some of the proofs could be extended to handle adaptivity.

A.5 Multi-party Quantum Computation

We define multi-party quantum computation by giving an ideal-model protocol for that task. Simply put, all players hand their inputs to the trusted party, who runs the desired circuit and hands back the outputs. Note that the only kind of cheating which is possible is that cheaters may choose their own input. In particular, cheaters cannot force the abortion of the protocol. One possible extension of this work is to consider protocols where cheaters may not compromise the correctness of the computation but might force the protocol to stop before completion.

A.6 Verifiable Quantum Secret Sharing

Providing a definition verifiable quantum secret sharing is trickier than it is for multi-party computing. The idea of the ideal protocol is simple. In the sharing phase, the dealer gives his secret system to the trusted party. In the reconstruction phase, the $\mathcal{TTP}$ sends the secret system to the reconstructor R.

[3] An adversary who corrupts players dynamically is called a mobile adversary, and protocols for handling such adversaries are called *pro-active*.

However, a problem arises because VQSS is a two phase task, and the formalism we established in the preceding sections only describes one-phase protocols, which have a simpler input/output behaviour. For example, if all we required of VQSS is that the reconstructor's output be the same as the dealer's input, we could simply have D send his secret system to R without violating the definition—a clear indication that such a definition would be insufficient. For the purposes of this paper, we adopt a simple modification of the definition of the preceding sections which allows us to describe VQSS: instead of giving all inputs to the parties at the beginning of the run of the protocol, some inputs are not given to the parties until the beginning of the reconstruction phase.

Specifically, two of the inputs are delayed. First, players learn the identity of the reconstructor R only at the beginning of the reconstruction phase (note that this doesn't stop the adversary from knowing R since the definition requires security for all adversaries and input sequences). Second, the adversary also receives a second auxiliary input $\mathcal{I}_{aux}^{(2)}$ at the beginning of the reconstruction. This allows us to capture any side information gained by the adversary during interactions which occur between the end of the sharing phase and the beginning of the reconstruction phase.

The ideal-model protocol we obtain is given in Protocol A.2. The definition of security we will use for this two-phase model is essentially the same as for the one-phase model. An input configuration ρ consists of player identities D and R, a secret system S and the two auxiliary inputs $\mathcal{I}_{aux}$ and $\mathcal{I}_{aux}^{(2)}$. We require that for all adversaries $\mathcal{A}_1$, there exists an adversary $\mathcal{A}_2$ such that for all sequences of input configurations $\{\rho_k\}_{k\in\mathbb{N}}$, the fidelity of the output of the real protocol to the output of the ideal protocol is exponentially close to 1.

B Further Preliminaries

B.1 Sharing Quantum Secrets and (No) Cloning

One of the fundamental theorems of quantum information theory is that an arbitrary quantum state cannot be cloned. In fact, one can say more: if there is a process with one input and two outputs, then if one of the outputs is a copy of the input, the other output *must be independent of the input.* We're not sure to whom this result is attributable but it has certainly become folklore.

Protocol A.1 (Multi-party Quantum Computation—Ideal Model).

Pre: All players agree on a quantum circuit U with n inputs and n outputs(for simplicity, assume that the i^{th} input and output correspond to player i).

Input: Each player gets an input system S_i (of known dimension, say p).

1. (**Input Sharing**) For each i, player i sends S_i to $\mathcal{TTP}$. If $\mathcal{TTP}$ does not receive anything, then he broadcasts "Player i is cheating" to all players. Otherwise, $\mathcal{TTP}$ broadcasts "Player i is OK."
2. (**Computation**) $\mathcal{TTP}$ evaluates the circuit U on the inputs S_i. For all i who cheated, $\mathcal{TTP}$ creates S_i in a known state (say $|0\rangle$).
3. (**Output**)
   (a) $\mathcal{TTP}$ sends i^{th} output to player i.
   (b) Player i outputs the system he receives from $\mathcal{TTP}$.

Protocol A.2 (Verifiable Quantum Secret Sharing—Ideal Model).

- **Sharing Phase:**
  1. **Inputs:** All players get D's identity. Dealer D gets a qupit S (i.e. a p-dimensional system, where p is a publicly agreed-upon integer).
     (Adversary also gets his auxiliary input $\mathcal{I}_{aux}$.)
  2. D sends the p-dimensional system S to $\mathcal{TTP}$. If D fails to send S, then $\mathcal{TTP}$ broadcasts "D is cheating" to all players. Otherwise, $\mathcal{TTP}$ broadcasts "OK".
- **Reconstruction Phase:**
  1. **Inputs:** All players get R's identity.
     (Adversary also gets his second auxiliary input $\mathcal{I}_{aux}^{(2)}$.)
  2. If D did not cheat in the sharing phase, $\mathcal{TTP}$ sends S to the receiver R.

Fact B.1 (No cloning, folklore). *Let $U : \mathcal{H}_m \otimes \mathcal{H}_W \longrightarrow \mathcal{H}_A \otimes \mathcal{H}_B$ ([4]) be a unitary transformation such that for all $|\psi\rangle \in \mathcal{H}_m$:*

$$U(|\psi\rangle \otimes |W\rangle) = |\psi\rangle \otimes |\varphi_{|\psi\rangle}\rangle$$

where $|W\rangle$ is some fixed auxiliary state (work bits). Then $|\varphi_{|\psi\rangle}\rangle$ does not depend on $|\psi\rangle$.

An important consequence of this was first pointed out by Cleve, Gottesman and Lo [14]: any quantum code is a scheme for sharing quantum secrets: A distance d code can correct $d-1$ erasures, and so access to any $n-d+1$ (uncorrupted) positions suffice to recover the encoded state; on the other hand, that means that any set of $d-1$ positions must reveal no information at all about the encoded state. That is, the density matrix of any $d-1$ positions is completely independent of the data.

Note that this phenomenon has no simple classical analogue: any position of a classical error-correcting code will leak information about the data unless the encoding process is randomized. This additional step is not necessary in the quantum setting since the randomness is "built in."

B.2 The Quantum Singleton Bound

As in classical coding theory, there is an easy relation btween erasure- and error-correction for quantum codes:

Fact B.2 (t-error correction and $2t$-erasure correction). *Suppose that a quantum code with n components, and dimension at least 2 can correct errors on any t positions. Then in fact $\mathcal{C}$ can correct* erasures *on any $2t$ positions.*

Note that this holds regardless of the dimensions of the individual components of the code. It also holds when the code in question is a "mixed state" code, i.e. some pure states are nonetheless encoded as mixed states by the encoding procedure.

[4] Note that in fact the mW system and the AB system are one and the same. The two labelings simply reflect a different partitioning of the system.

Now a corollary of the no cloning theorem of the previous section is that no quantum code with n components can withstand the erasure of $\lceil n/2 \rceil$ components. If it could, then one could always separate the codeword into two halves and reconstruct a copy of the encoded data with each half, yielding a clone of the encoded data. By the equivalence of t-error-correction and $2t$-erasure-correction, this means that *there is no quantum code that can correct errors on any* $\lceil n/4 \rceil$ *positions.* This is a special case of the quantum Singleton bound, also called the Knill-Laflamme bound.

B.3 Tools from Fault-Tolerant Quantum Computing

In our proposed solution, we also use techniques developed for fault-tolerant quantum computing (FTQC). The challenge of FTQC is to tolerate *non-malicious* faults occurring within a single computer. One assumes that at every stage in the computation, every qubit has some probability p of suffering a random error, i.e. of becoming completely scrambled (this corresponds to the classical notion of random bit flips occurring during a computation). Moreover, errors are assumed to occur *independently* of each other and of the data in the computation. See Section 1.1 for a discussion of the difference between FTQC and MPQC. In this section, we review a number of useful results from FTQC. These come from [27, 3, 21].

Universal Sets of Gates The usual technique behind fault-tolerant computing (both classical and quantum) is to design procedures for applying one of a small number of gates to logical (i.e. encoded) values, without having to actually decode the values and then re-encode them. That is, given the encoding state $|\psi\rangle$, we want a simple procedure which returns the encoding of state $U|\psi\rangle$.

Thus, it is useful to find a small set of gates which is *universal*, i.e which suffices to implement any desired function[5]. One can then simply design fault-tolerant procedures for implementing these gates, and compose them to obtain a fault-tolerant procedure for any particular function.

For qupits of prime dimension p, [3] showed that the following set of gates is universal:

1. Generalized NOT (a.k.a. X): $\forall\ c \in F$, $|a\rangle \longmapsto |a+c\rangle$,
2. Generalized CNOT (Controlled Addition): $|a, b\rangle \longmapsto |a, a+b\rangle$,
3. Swap $|a\rangle|b\rangle \longmapsto |b\rangle|a\rangle$,
4. Multiplication gate: $0 \neq c \in F$: $|a\rangle \longmapsto |ac\rangle$,
5. Phase Shift (a.k.a. Z): $\forall c \in F\ |a\rangle \longmapsto w^{ca}|a\rangle$,
6. Generalized Hadamard (Fourier Transform): $|a\rangle \longmapsto \frac{1}{\sqrt{p}} \sum_{b \in F} w^{rab}|b\rangle, \forall 0 < r < p.$
7. Generalized Toffoli: $|a\rangle|b\rangle|c\rangle \longmapsto |a\rangle|b\rangle|c+ab\rangle$,

Beyond these, in order to simulate arbitrary quantum circuits one should also be able to introduce qupits in some known state (say $|0\rangle$), as well as to discard qupits. Note that these are sufficient for simulating measurements, since one can simply apply a controlled-not with a state $|0\rangle$ as the target and then discard that target.

[5] In fact, it is impossible to find a finite set which can implement any unitary operation perfectly. However, one can approximate any unitary operation on a constant number of qubits to accuracy ϵ using $O(poly \log \frac{1}{\epsilon})$ gates from a "universal" set, i.e. one which generates a group which is dense in the space of all unitary operators.

Transversal Operations Fortunately, several of these gates can be applied *transversally*, that is using only "qupit-wise" operations. These are important since they correspond to operations performed locally by the players in a multi-party protocol, if each player shares has one component of an encoded state.

For example: in any CSS code, the linear gate $|a, b\rangle \longmapsto |a, a+cb\rangle$ can be applied to two encoded qupits by applying the same gate "qupit-wise" to the two codewords. For any CSS code, the gates 1 through 5 from the set above can be implemented transversally [27, 3].

Remark 1. Another operation which can almost be performed transversally is measurement in the computational basis. The encoding of a classical state $|s\rangle$ in a CSS code is the equal superposition of all the words in some particular coset of $V_0 = W^{\perp}$ within V. Thus, measuring all the qupits of the encoding of $|s\rangle$ will always yield a codeword from that coset. Similarly, measuring all the qupits of the encoding of $\sum_s \alpha_s|s\rangle$ will yield a word from the coset corresponding to s with probability $|\alpha_s|^2$. This operation is not quite transversal since after the qupit-wise measurement, the classical information must be gathered together in order to extract the measurement result. Nonetheless, the quantum part of the processing is transversal, and this will be good enough for our purposes.

Transversal Fourier Transforms and the Dual Code In general, applying the Fourier transform transversally to a codeword from a CSS code $\mathcal{C}$ does not yield a word from that code. Instead, one obtains a word from the "dual code" $\tilde{\mathcal{C}}$. If $\mathcal{C}$ is defined by the classical codes V and W, then $\tilde{\mathcal{C}}$ is the CSS code obtained using the codes W and V. A natural choice of encoding for the dual code yields the following relation:

$$\mathcal{F}^{\otimes n}\mathcal{E}_{\mathcal{C}}|\psi\rangle = \mathcal{E}_{\mathcal{C}^{\delta'}}\left(\mathcal{F}|\psi\rangle\right)$$

where $\mathcal{E}_{\mathcal{C}}$ and $\mathcal{E}_{\mathcal{C}^{\delta'}}$ are the encoding operators for $\mathcal{C}$ and $\mathcal{C}^{\delta'}$ respectively.

For polynomial codes of degree δ, recall that there is related degree $\delta' = n - \delta - 1$. As one can observe from the dual codes $W^{\delta'}, W_0^{\delta'}$, the dual code $\widetilde{\mathcal{C}^{\delta}}$ is a "mangled" version of the code $\mathcal{C}^{\delta'}$. In fact, by scaling each Fourier transform with the (non-zero) factor d_i, one obtains:

$$\mathcal{F}^{\mathbf{d}}\mathcal{E}_{\mathcal{C}^{\delta}}|\psi\rangle = \mathcal{E}_{\mathcal{C}^{\delta'}}\left(\mathcal{F}|\psi\rangle\right)$$

Note that when n is exactly $2\delta + 1$, the codes $\mathcal{C}^{\delta}$ and $\mathcal{C}^{\delta'}$ are the same, and so the Fourier transform on encoded data can in fact be applied transversally: $\mathcal{F}^{\mathbf{d}}\mathcal{E}_{\mathcal{C}^{\delta}}|\psi\rangle = \mathcal{E}_{\mathcal{C}^{\delta}}\left(\mathcal{F}|\psi\rangle\right)$.

Transversal Reductions to *Degree Reduction* for $\delta < n/3$ As mentioned above, the only operations that cannot, in general, be performed transversally on Reed-Solomon codes are the Fourier transform and Toffoli gate. However, when δ is less than $n/3$, [3] reduces both of them to the problem of *degree reduction*, which involves mapping the encoding of $|\psi\rangle$ under the dual code $\mathcal{C}_{\delta'}$ to the encoding of $|\psi\rangle$ under the original code $\mathcal{C}_{\delta}$.

For the Fourier transform, the reduction is obvious: we showed above that by performing (scaled) Fourier transforms transversally to $\mathcal{E}_{\mathcal{C}_{\delta}}|\psi\rangle$, one obtains $\mathcal{E}_{\mathcal{C}_{\delta'}}\left(\mathcal{F}|\psi\rangle\right)$. Thus, performing degree reduction would produce $\mathcal{E}_{\mathcal{C}_{\delta}}\left(\mathcal{F}|\psi\rangle\right)$, which is the desired result.

For the Toffoli gate, note that $\delta < n/3$ implies that $\delta' = n - \delta - 1$ is at least 2δ. The underlying idea is simple: suppose we have three polynomials p, q, r of degree such that $p(0) = a, q(0) = b$ and $r(0) = c$. Take the polynomial r' given by $r'(i) = r(i) + p(i)q(i)$ for all $i = 1, ..., n$. First, note that if p, q have degree at most δ and r has degree at most $\delta' \geq 2\delta$, then $\deg(r') < \delta'$. Moreover, if p, q, r are *random* polynomials subject to the above constraints, then p, q, r' will also form a random triple of polynomials, which interpolate to the values $a, b, c + ab$.

To map this to a procedure for implementing the Toffoli gate, suppose that we have the encodings of $|a\rangle$ and $|b\rangle$ using the code $\mathcal{C}^{\delta}$. Suppose that we also have the encoding of $|c\rangle$ using the related code $\mathcal{C}^{\delta'}$. By applying the Toffoli gate qupit-wise, we obtain the encoding of $c+ab$ under the related code:

$$\mathcal{E}_{\mathcal{C}^\delta}|a\rangle\mathcal{E}_{\mathcal{C}^\delta}|b\rangle\mathcal{E}_{\mathcal{C}^{\delta'}}|c\rangle \longmapsto \mathcal{E}_{\mathcal{C}^\delta}|a\rangle\mathcal{E}_{\mathcal{C}^\delta}|b\rangle\mathcal{E}_{\mathcal{C}^{\delta'}}|c+ab\rangle$$

Thus, to implement the Toffoli gate fault-tolerantly it is sufficient to have an implementation of the two maps $\mathcal{E}_{\mathcal{C}^\delta}|\psi\rangle \longmapsto \mathcal{E}_{\mathcal{C}^{\delta'}}|\psi\rangle$ and $\mathcal{E}_{\mathcal{C}^{\delta'}}|\psi\rangle \longmapsto \mathcal{E}_{\mathcal{C}^\delta}|\psi\rangle$. Note that this is equivalent to having a procedure for just one map $\mathcal{E}_{\mathcal{C}^{\delta'}}|\phi\rangle \longmapsto \mathcal{E}_{\mathcal{C}^\delta}|\phi\rangle$, since one can simply apply the Fourier transform first and its inverse afterwards to reverse the direction.

Implementing Degree Reduction The circuit we use for degree reduction is due to Gottesman and Bennett [20] (and is related to [21]), and is much more efficient than the original one proposed in [3]. Begin with the state to be transformed (call this system $\mathcal{H}_1$) and an ancilla in state $\mathcal{E}_{\mathcal{C}^\delta}|0\rangle$ (called $\mathcal{H}_2$).

1. Apply controlled addition from $\mathcal{H}_1$ to $\mathcal{H}_2$.
2. Apply the scaled Fourier transform transversally to $\mathcal{H}_1$.
3. Measure $\mathcal{H}_1$ in the computational basis, obtaining b.
4. Apply a conditional phase shift with scaling factor $-b$ to $\mathcal{H}_2$.

The effect of this on the basis state $\mathcal{E}_{\mathcal{C}}|a\rangle$ (for $a \in \mathbb{Z}_p$) is:

$$\begin{aligned}\mathcal{E}_{\mathcal{C}}|a\rangle\mathcal{E}_{\tilde{\mathcal{C}}}|0\rangle \mapsto \mathcal{E}_{\mathcal{C}}|a\rangle\mathcal{E}_{\tilde{\mathcal{C}}}|a\rangle &\mapsto \sum_b \omega^{ab}\mathcal{E}_{\mathcal{C}}|b\rangle\mathcal{E}_{\tilde{\mathcal{C}}}|a\rangle \\ &\mapsto \omega^{ab}\mathcal{E}_{\tilde{\mathcal{C}}}|a\rangle(\text{with } b \text{ known}) \mapsto \mathcal{E}_{\tilde{\mathcal{C}}}|a\rangle\end{aligned}$$

This procedure in fact works for arbitrary linear combinations (intuitively, this is because the measurement result b yields no information about a).

Note that this entire procedure can be performed transversally except for the measurement step. However, as noted above (Remark 1), measurement requires only classical communication between the components (namely, each component is measured and the classical decoding algorithm for the code $V^{\delta'}$ is applied to the result).

C More on Neighborhoods of Quantum Codes

Note that the notions $N_B(\mathcal{C})$ and $ST_B(\mathcal{C})$ make sense for any subspace $\mathcal{C}$ of $\mathcal{H}$. On the one hand, we always have $N_B(\mathcal{C}) \subseteq ST_B(\mathcal{C})$ since local operations do not affect the density matrix of other components. If we restrict our attention to pure states, then $N_B(\mathcal{C})$ and $ST_B(\mathcal{C})$ are in fact identical. Specifically, define:

$$\begin{aligned}N_B^{pure}(\mathcal{C}) &= \Big\{|\psi\rangle \in \mathcal{H} : \exists|\phi\rangle \in \mathcal{C}, \exists U \text{ unitary, acting only on } \mathcal{H}_B \\ &\qquad \text{such that } |\psi\rangle = (I_A \otimes U)|\phi\rangle\Big\} \\ ST_B^{pure}(\mathcal{C}) &= \Big\{|\psi\rangle \in \mathcal{H} : \exists|\phi\rangle \in \mathcal{C}, \mathrm{Tr}_B(|\psi\rangle\langle\psi|) = \mathrm{Tr}_B(|\phi\rangle\langle\phi|)\Big\}\end{aligned}$$

Proposition C.1. *For any subspace $\mathcal{C}$: $N_B^{pure}(\mathcal{C}) = ST_B^{pure}(\mathcal{C})$*

Proof: We must only prove $N_B^{pure}(\mathcal{C}) \supseteq ST_B^{pure}(\mathcal{C})$, since the other inclusion is trivial. Take any state $|\psi\rangle \in ST_B^{pure}(\mathcal{C})$. Let $|\phi\rangle$ be the corresponding state in $\mathcal{C}$ and let $\rho = \mathrm{Tr}_B(|\psi\rangle\langle\psi|) = \mathrm{Tr}_B(|\phi\rangle\langle\phi|)$. We can write $\rho = \sum_i p_i |a_i\rangle\langle a_i|$ with $p_i > 0$, $\sum_i p_i = 1$, and the vectors $|a_i\rangle$ orthonormal.

By the Schmidt decomposition, we know that we can write

$$|\psi\rangle = \sum_i \sqrt{p_i}|a_i\rangle \otimes |b_i\rangle$$

with the vectors $|b_i\rangle$ orthonormal. Similarly, there is some other set of orthonormal vectors $|b'_i\rangle$ such that we can write

$$|\phi\rangle = \sum_i \sqrt{p_i}|a_i\rangle \otimes |b'_i\rangle$$

Now consider any unitary matrix U on $\mathcal{H}_B$ which maps $|b'_i\rangle$ to $|b_i\rangle$. Such a matrix always exists since the sets of vectors are orthonormal. Then we have $|\psi\rangle = (I_A \otimes U_B)|\phi\rangle$ as desired. □

This equality does not hold once we relax our definition and consider mixed states. Namely:

Proposition C.2. *There exist subspaces $\mathcal{C}$ for which $N_B(\mathcal{C}) \subsetneq ST_B(\mathcal{C})$.*

Proof: Many quantum codes have this property, for an appropriate partition of the code word into parts A and B. Take $\mathcal{C}$ to be a quantum RS code with $n = 2\delta + 1$. Encode 1/2 of an EPR pair. Now get ρ by appending the other half of the EPR pair to the end of the codeword (say there is space left over in position n, for example). On one hand, ρ is clearly in ST_B as long as B includes position n. However, it is not in $N_B(\mathcal{C})$ since the only state "in" $\mathcal{C}$ which has the same trace as ρ on A is a mixed state. □

For the case of CSS codes, we can additionally define $\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}$. Again, there is a trivial inclusion: $ST_B(\mathcal{C}) \subseteq \mathcal{C}_B$. This inclusion also holds when we restrict our attention to pure states. However, the inclusion is strict, even for pure states:

Proposition C.3. *There exist subspaces $\mathcal{C}$ for which $ST_B(\mathcal{C}) \subsetneq \mathcal{C}_B$.*

Proof: Again, consider the quantum RS code with $n = 2\delta + 1$. Take $A = \{1, ..., \delta + 1\}$ and $B = \{n - \delta + 1, ..., n\}$. Both V_B and W_B cover the entire space $\mathbb{Z}_p^n$, so in fact $\mathcal{C}_B$ is the entire Hilbert space. However, any state ρ in $ST_B(\mathcal{C})$ must yield $\rho' \otimes I_{\{2,...,\delta+1\}}$ when the interpolation operator is applied to the positions of ρ in A. Thus, not all states, pure or mixed, are in $ST_B(\mathcal{C})$. □

It should be noted that neither $N_B(\mathcal{C})$ nor $ST_B(\mathcal{C})$ are subspaces. Moreover, for CSS codes, $\mathcal{C}_B$ is the subspace generated by the vectors in $N_B(\mathcal{C})$.

C.1 Proofs from Section 2

Recall that given a state ρ, testing if ρ is in $V_B^{(q)}$ is easily described: For each element of (a basis of) the dual space $V_B^{\perp}$, we measure the corresponding linear combination of the qupits of ρ in the computational basis, and check that it is 0. Recall that the vectors of the dual space $V_B^{(q)}$ have support only on A (since arbitrary changes to positions in B should not affect whether or not a word is in V_B), and so one need not have access to the components in A in order to perform the measurement. Similarly, to check if ρ is in $\mathcal{F}^{\otimes n}W_B^{(q)}$, we rotate into the Fourier basis and measure the linear combinations corresponding to a basis of $W_B^\perp$.

Note that since $V_B^\perp \subseteq V^\perp$ and $W_B^\perp \subseteq W^\perp$, and since measuring the V-syndrome in the computational basis commutes with measuring the W-syndrome in the Fourier basis, we know that the following four measurements commute:

1. V_B-syndrome in the computational basis
2. V-syndrome in the computational basis
3. W_B-syndrome in the Fourier basis
4. W-syndrome in the Fourier basis

Proof (of Lemma 2.1): As was just mentioned, to check if ρ is in $\mathcal{C}_B$, we measure the V_B-syndrome in the computational basis and the W_B-syndrome in the Fourier basis. But by the remarks above, the distribution on this outcome measurement will not change if we first measure the V and W syndromes, i.e. if we first make a measurement which projects ρ into one of the subspaces $E_j\mathcal{C}$ (i.e. ρ maps to $\rho' = P_j \rho P_j$ with probability $\mathrm{Tr}\,(P_j\rho)$, where P_j is the projector for the space $E_j\mathcal{C}$).

The new state ρ' lies completely in one of the spaces E_j. However, $E_j\mathcal{C}$ is either contained in $\mathcal{C}_B$ (if there is an operator equivalent to E_j which acts only on B) or *orthogonal* to $\mathcal{C}_B$ (if no such operator exists).

Thus, for ρ to have fidelity 1 with $\mathcal{C}_B$, it must be that $\mathrm{Tr}\,(P_j\rho) = 0$ for all E_j which act on more than B. Hence ρ is a mixture of states $|\psi_i\rangle$ each of which is a linear combination of elements of the spaces $\{E_j\mathcal{C}\}$, where E_j acts only on B. □

Proof (of Proposition 2.2): Consider a particular basis state $E_j\mathcal{E}|a\rangle$. The decoding operator $\mathcal{D}$ will produce the state $|a\rangle|j\rangle$, since errors of weight at most t can be identified uniquely. The ideal operator $\mathcal{R}^I$ will extract the encoded state $|a\rangle$. Without loss of generality, the ideal recovery operator will replace $|a\rangle$ with $|0\rangle$, the final output $|a\rangle \otimes E_j\mathcal{E}|0\rangle$.

In both cases, the output can be written as $|a\rangle$ tensored with some ancilla whose state depends only on the syndrome j (and which identifies j uniquely). Once that state is traced out, the outputs of both operators will be identical. Another way to see this is that the ideal operator can simulate the real operator: one can go from the output of the ideal operator to that of the real operator by applying a transformation which only affects the ancilla. For a state ρ expressed as in Lemma 2.1, the final outcome will be $\rho' = \sum_{ij} p_i |c_{ij}|^2 |\phi_{ij}\rangle\langle\phi_{ij}|$. □

D Proofs of Security (from Sections 3 and 5)

D.1 Subspace Projection

Before proving the secuiry of the VQSS protocol, we consider a protocol for a simpler task that we call *subspace projection*, which illustrates the key ideas in the VQSS protocol. Namely, we first modify a classical protocol of [12] so that the dealer does not have to remember the random bits he used in sharing his secret. Second, we apply this protocol both in the computational and Fourier bases. We use a "quantum-to-classical" argument to show that this garantees that the joint state shared by the players satisfies condition (3) from the discussion on neighborhoods, i.e. that the joint state passes certain local checks in both bases. For concreteness, let W be the RS code V^δ, where $n = 4t + 1$ and $\delta = 2t$.

Let $\mathcal{H}_0, \ldots, \mathcal{H}_k$ be separate quantum systems consisting of n qupits each, and let $\mathcal{H} = \mathcal{H}_0 \otimes \cdots \otimes \mathcal{H}_k$. Say the dealer prepares $\mathcal{H}$ in some state and gives the ith qupit of each subsystem $\mathcal{H}_j$ to player i. He wants to prove to the group that in fact the fidelity of $\mathcal{H}_0$ to the space $W^{(q)}$ is close to 1 [6], without revealing any information beyond that to the other players. What we achieve in this first step is not quite that strong: at the end of the protocol, there will be a publicly known set B of "apparent cheaters" such that the shares of the honest players not in B will all agree with W in the computational basis, i.e. will have high fidelity to the space $W^{(q)}_{B \cup C}$.

Protocol D.1 (Subspace projection).

1. **Sharing** The dealer D prepares $\mathcal{H}_0$ as any state (pure or mixed) in $W^{(q)}$ and distributes it to the players. He then prepares $\mathcal{H}_1, \ldots, \mathcal{H}_k$ in the equal superposition of $\sum_{\mathbf{w} \in W} |\mathbf{w}\rangle$, and distributes those to the players also.
2. **Verification** Using classical VSS, every player commits to k field elements picked uniformly at random. These commitments are then opened and their sum is taken to obtain k field elements $b_1, \ldots, b_k$ (these are completely unpredictable to the dealer, even if he is cheating).
3. For $\ell = 1, \ldots, k$, players apply the linear operation $(x, y) \mapsto (x, y + b_\ell x)$ to the subsystems $\mathcal{H}_0$ and $\mathcal{H}_\ell$. All players then measure their shares of $\mathcal{H}_1, \ldots, \mathcal{H}_k$ in the computational basis and broadcast the result.
4. Each of the broadcasted words $\mathbf{w}_1, \ldots, \mathbf{w}_k$ is decoded using classical error-correction of the code W: for each $\mathbf{w}_\ell$, we obtain either that it was at distance more than t from a word in W or we obtain an error vector with support $B_\ell \in \{1, ..., n\}$ of size less than t on which $\mathbf{w}_\ell$ differs from a word in W.

   The dealer is rejected if any of the broadcasted words was at distance more than t or if $B = \bigcup_{\ell=1}^{k} B_\ell$ has size greater than t. Otherwise, the dealer is accepted.

Lemma D.1. *Let $\tilde{B} = B \cup C$. At the end of the protocol above, the fidelity of the system to the statement "either $\mathcal{H}_0$ is in $(W_{\tilde{B}})^{(q)}$ or the dealer has been rejected" is exponentially close to 1 in k.*

To prove this, we give a "quantum to classical" reduction, as in [22].

Lemma D.2. *Consider the subspace projection protocol above. Then the behavior of the protocol is the same in each of the two following experiments:*

Experiment 1 *at the end of the* whole protocol*, all honest players measure their shares of $\mathcal{H}_0$ in the computational basis,* or

Experiment 2 *at the end of the* sharing phase*, all honest players measure their shares of $\mathcal{H}_0$ and $\mathcal{H}_1$ in the computational basis, and then run the verification phase.*

Moreover, the distribution on the results of the measurement of $\mathcal{H}_0$ is the same in both cases.

Proof: The actions of the honest players on their shares in the original protocol can be seen as the composition of k super-operators, each of which is comprised of two operations: a controlled-addition gate from $\mathcal{H}_0$ to $\mathcal{H}_\ell$ followed by measurement of $\mathcal{H}_\ell$. Denote the controlled-addition gate by $(c\text{-}X^b)_\ell$, where b is the scaling factor for the controlled-addition. Second, denote measurement of $\mathcal{H}_\ell$ in the computational basis by $\mathcal{M}_\ell$.

[6] It would be desirable to be able to prove that the fidelity is in fact exactly 1. This remains an interesting open question. This corresponds to the classical difference between zero-error and small-error protocols.

Consider what happens in the ℓ^{th} verification step in Experiment 1. Because the controlled-addition gate is a permutation of the basis states of the computational basis, measuring the systems in that basis *before* the gate is applied will not change the outcome of measurements made *after* the gate is applied. Thus we can write $\mathcal{M}_\ell\mathcal{M}_0(c\text{-}X^{b_\ell})_\ell = \mathcal{M}_\ell\mathcal{M}_0(c\text{-}X^{b_\ell})_\ell\mathcal{M}_\ell\mathcal{M}_0$, and the distribution of the measurements made after the gate is applied will not change.

But now notice that measuring the system $\mathcal{M}_0$ afterwards is completely redundant. Because the controlled-addition gate does not change the first component of any basis vectors, measuring $\mathcal{M}_0$ after the application of the gate will yield the same result as measuring it before. Hence, we can write $\mathcal{M}_\ell\mathcal{M}_0(c\text{-}X^{b_\ell})_\ell = \mathcal{M}_\ell(c\text{-}X^{b_\ell})_\ell\mathcal{M}_\ell\mathcal{M}_0$. However, this is exactly the sequence of operations performed by honest players in Experiment 2: first they measure both systems, then apply the addition gate and measure the target.

Thus, the measurement outcomes will be the same in both experiments will be the same. Moreover, the cheaters can see no difference between the two experiments, and so their behavior will not change. □

In other words, we can imagine two situations. In the first one, just after the sharing phase of the protocol, an outsider comes in and secretly measures honest players' shares of $\mathcal{H}_0, \ldots, \mathcal{H}_k$ in the computational basis. In the second, the outsider performs this secret measurement *after* the protocol is completed. The statement is that exactly when he makes his measurement will not change the behavior of the protocol.

But recall that our original statement of the correctness of the protocol is that, at the end of the protocol, either the dealer has been caught or the shares of players are in $W_{\tilde{B}}^{(q)}$. Since fidelity to $W_{\tilde{B}}^{(q)}$ is the same as the probability that measuring in the computational basis gives a word in $W_{\tilde{B}}$ (i.e. agrees with W when truncated to positions neither in B nor C), Lemma D.2 allows us to restrict ourselves to thinking about situations in which the shares of the systems $\mathcal{H}_0, \ldots, \mathcal{H}_k$ sent to honest players were in fact classical states.

Now consider the classical protocol corresponding to the subspace projection protocol: the dealer distributes $k+1$ codewords $\mathbf{w}_0, \ldots, \mathbf{w}_k$. At each step, a random multiple of $\mathbf{w}_0$ is added to one of the other codewords and the result is broadcast. At the end, players compute B as above and decide whether or not to reject the dealer. (This is the blob protocol of [12], modified so as not to require the involvement of the dealer beyond the sharing stage).

Fact D.3 (Soundness of Modified Blobs from [12]). *At the end of classical protocol, let A be the set of honest players not in B. The event "either the players in A have consistent shares or the dealer was caught" occurs with probability at least* $1 - 2^{n-k}$, even when the adversary is adaptive.

D.2 Dual Subspace Projection

Consider a "dual" version of the subspace projection protocol above. It is the same as the original protocol, with three changes:

1. Before proceeding to the verification phase all players apply the Fourier transform to all their shares.
2. At the end all players apply the inverse Fourier transform to their shares of $\mathcal{H}_0$.
3. (When D is honest) D prepares the ancillas $\mathcal{H}_1, ..., \mathcal{H}_k$ as a superposition over all words from the dual code $W^\perp$ (i.e. $\sum_{\mathbf{w}\in W^\perp} |\mathbf{w}\rangle$).

Now the state $\sum_{\mathbf{w}\in W^\perp}|\mathbf{w}\rangle$ is the image of $\sum_{\mathbf{w}\in W}|\mathbf{w}\rangle$ under transversal Fourier transforms. Thus, we can use the same analysis as in the previous section. At the end of this "dual" protocol, the fidelity of the system to the statement "either the dealer is caught or $\mathcal{H}_0$ is in the space $\mathcal{F}^{\otimes n}W_{\tilde{B}}^{(q)}$" is high. But recall that conjugating by Fourier rotations maps linear gates to linear gates (see Section 1.3). In particular, controlled addition gates simply have their direction reversed, i.e. source and target are swapped. Thus, the modifications to the original subspace projection protocol can be restated as follows:

1. the controlled addition gates are performed *from $\mathcal{H}_\ell$ to $\mathcal{H}_0$*;
2. the measurements are made in the rotated (Fourier) basis;
3. (When D is honest) D prepares the ancillas $\mathcal{H}_1, ..., \mathcal{H}_k$ as a superposition over all words from the dual code $W^\perp$ (i.e. $\sum_{\mathbf{w}\in W^\perp}|\mathbf{w}\rangle$).

D.3 Proofs for VQSS

To prove the secuirty of the VQSS protocol, first note that classically, 2-GOOD trees guarantee a unique shared secret, and that secret can be reconstructed efficiently using decoding.

Fact D.4. *Suppose that a sharing is 2-GOOD. If all players broadcast their shares, then the same value a will always be reconstructed for the root of the tree, regardless of the cheaters' values.*

Note that the protocols (and proofs) of [12] used this fact implicitly, but did not use the recovery algorithm as stated here. Instead, they required players to remember what shares they had distributed to other players.

D.3.1 Classical VSS

We give a modified version of the VSS protocol of [12]. The main difference is that the original protocol required a dealer to remember the values of the shares sent in the first phase, and cooperate later on during the verification phase. However, this does not generalize well to the quantum world, and so we compensate by exploiting the efficient decodability of Reed-Solomon codes. The protocol is given in Protocol D.2. Note that as before, the error-correcting code we use is V^δ, where $n = 4t+1$ and $\delta = 2t$.

Remark 2. In the description of the protocol (and subsequent protocols), we assume for simplicity that there is a source of public randomness. This is not a problem in our setting as good random bits can be generated using classical VSS protocols, and it simplifies the analysis of the protocols. However, it is not necessary (one can have players take turns broadcasting challenges, as in [12, 26]).

The correctness and soundness of this protocol are stated here. They follow from the properties of 2-GOOD trees and from cut-and-choose analysis.

Fact D.5. *If D is honest, he will pass the protocol with probability 1, and the shares $\mathbf{v}_{0,i}(j)$ will form a 2-GOOD tree which interpolates to the original input a.*

Fact D.6. *With probability $1-2^{\Omega(k)}$, either the shares $\mathbf{v}_{0,i}(j)$ form a 2-GOOD tree or the dealer is caught during the protocol.*

Protocol D.2 (Modified Classical VSS **from [12]).** The dealer D has input $a \in F$.

- **Sharing:**
  1. D picks a random codeword $\mathbf{v}_0 \in V$ such that $\mathbf{v}_0$ interpolates to a. D also picks k random codewords $\mathbf{v}_1, ..., \mathbf{v}_k \in V$ (i.e. k sharings of random values).
  2. D gives player i the i^{th} component of each of these vectors: $\mathbf{v}_\ell(i)$ for $\ell = 0, ..., k$.
  3. Player i shares each of these values with random vectors $\mathbf{v}_{0,i}, ..., \mathbf{v}_{k,i}$ which interpolate to $\mathbf{v}_0(i), ..., \mathbf{v}_k(i)$, respectively. He sends the values $\mathbf{v}_{\ell,i}(j)$ to player j (for $\ell = 0, ..., k$).
- **Verification:** Get k previously unknown public random values $b_1, ..., b_k$. For $\ell = 1, ..., k$:
  1. For all i, player j broadcasts $\mathbf{v}_{\ell,i}(j) + b_\ell \mathbf{v}_{0,i}(j)$.
     (i.e. player j broadcasts his share of $\mathbf{v}_{\ell,i} + b_\ell \mathbf{v}_{0,i}$).
  2. For each $i \in \{1, ..., n\}$, players update the set B_i based on the broadcast values, as in the subspace projection protocol. If there are too many errors, then they add i to the global set B.
  3. Furthermore, players do the same at the branch level: for all $i \notin B$, there is an interpolated value a_i which corresponds to the decoded codeword from the previous step. Players also decode the codeword $(a_1, ..., a_n)$ and update B accordingly (i.e. by adding any positions where errors occur to B).
- The dealer is disqualified if B is ever larger than t.
- If the dealer passes, the values $\mathbf{v}_{0,i}(j)$ are taken to be the shares of the dealer's secret.
- **Reconstruction:**
  1. Player j broadcasts his shares $\mathbf{v}_{0,i}(j)$ for all i.
  2. Let T be the tree defined by these values. All players output the value given by $\mathsf{Recover}(T, V, B, B_1, ..., B_n)$.

D.3.2 Proof of Soundness

Proof (of Lemma 3.1): The proof of this lemma follows the ideas outlined in the proof of soundness for the subspace projection protocol. First, a quantum-to-classical reduction allows us to use the soundness of the modified classical protocol from Section D.3.1: this gives us that at the end of Step 1, either D would get caught or all the systems $S_{\ell,0}$ would yield 2-GOOD$_V$ trees if measured in the computational basis. After applying the Fourier transformations in Step 2, all the systems will be 2-GOOD$_V$ in the Fourier basis. Subsequent application of linear gates will not change that, since they correspond to linear gates in the Fourier basis. Finally, applying a second quantum-to-classical reduction shows that at the end of Step 3, the system $S_{0,0}$ will be 2-GOOD$_W$ in the computational basis. Since it is also 2-GOOD$_V$ in the Fourier basis, the final rotation in Step 4 will leave it 2-GOOD$_V$ in the computational basis and 2-GOOD$_W$ in the Fourier basis. □

Now, let $\mathcal{E}$ denote the operator used to encode a state using $\mathcal{C}$. Let J be a set of indices j such that the error operators $\{E_j\}_{j \in J}$ run over all the syndromes of the code $\mathcal{C}$ (i.e. J contains one representative from each equivalence class of error operators, and the spaces $\{E_j \mathcal{C}\}_{j \in J}$ are orthogonal and span $\mathbb{C}^{p^n}$). Note that $|J| = p^{n-1}$ since the code is 1-dimensional.

Proposition D.7 (Characterizing 2-GOOD trees). *The space of trees of qupits which are 2-*GOOD$_V$ *in the computational basis and 2-*GOOD$_W$ *in the Fourier basis is spanned by the states*

$\left\{E_{j_1}^{(1)} \cdots E_{j_n}^{(n)} \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E}|a\rangle\right\}$ *where (i) E_{j_0} (or something in its equivalence class) acts only on B and (ii) for each $i \notin B$, E_{j_i} (or something in its equivalence class) acts only on $B_i \cup C$. (Recall that for i corresponding to honest players not in B, we have $B_i \subseteq C$ and so in those cases the condition is that E_{j_i} act only on C.)*

Note that in the case of the basis vectors of the previous proposition, there is no entanglement between the data and the errors, since the data is a pure state (in fact, we can also think of the errors as being described by a pure state $|j_0, ..., j_n\rangle$). However, one can get arbitrary superpositions of these basis vectors and so in general there will be not only correlation, but indeed entanglement between the data and the errors.

To prove Proposition D.7, we will need:

Fact D.8. *The following set is an orthonormal basis of p^{n^2}-dimensional Hilbert space $\mathbb{C}^{p^{n^2}}$ (where p is the size of F):*

$$\left\{E_{j_1}^{(1)} \cdots E_{j_n}^{(n)} \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E}|a\rangle \quad : \quad j_0, ..., j_n \in J, a \in F\right\}$$

where the superscript $^{(i)}$ on E_{j_i} indicates that it acts on the i^{th} block of n qupits.

Proof: This is just writing the space in terms of cosets of a concatenated code.□

Proof (of Proposition D.7): Given any state of n^2 qupits, we can write it as a mixture of linear combinations of basis vectors from the basis in the previous discussion (Fact D.8). Now for any one of these basis states given by $j_0, ..., j_n$ and a, it will pass a test of 2-GOOD-ness in both bases if and only if the conditions of the proposition are satisfied: j_0 should be the syndrome of some error which acts only on B and each j_i should be equivalent to an error on $B_i \cap C$. Thus, any state which passes the test with probability 1 can in fact be written only in terms of those basis vectors which pass the test. □

Proof (of Lemma 3.2): Both the decoding and recovery operators produce an output qubit as well as an ancilla. We show that there is a unitary map which can be applied to the ancilla of the interpolation operator so that the joint state of the output and the ancilla are the same as when the decoding operator is applied. For simplicity, we assume that the interpolation circuit extracts the encoded state and replaces it with an encoding of $|0\rangle$, i.e. it maps $\mathcal{E}|a\rangle \longmapsto |a\rangle \otimes \mathcal{E}|0\rangle$.

It is sufficient to prove this for some basis of the space of 2-GOOD trees; the rest follows by linearity. The natural basis is given by Proposition D.7. Consider a basis vector $E_{j_1}^{(1)} \cdots E_{j_n}^{(n)} \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E}|a\rangle$ which satisfies the conditions of Proposition D.7.

Effect of ideal recovery Let I be the set of $n - 2t$ indices not in either B or C, and suppose for simplicity that $I = \{1, ..., n - 2t\}$ (the same argument works regardless of the particular values in I). Applying the ideal recovery operator to the branches in I, we obtain $n - 2t$ encodings of $|0\rangle$ with errors $j_1, ..., j_{n-2t}$, and an encoding of $|a\rangle$ whose first $n - 2t$ positions are untouched and whose last $2t$ positions are themselves encoded and possibly arbitrarily corrupted. This can be written:

$$(E_{j_1}\mathcal{E}|0\rangle) \otimes \cdots \otimes \left(E_{j_{n-2t}}\mathcal{E}|0\rangle\right) \ \otimes \ E_{j_{n-2t+1}}^{(n-2t+1)} \cdots E_{j_n}^{(n)} \left(\mathbb{I}^{\otimes(n-2t)} \mathcal{E}^{\otimes 2t}\right) E_{j_0}\mathcal{E}|a\rangle$$

where $\mathbb{I}$ is the identity. Applying ideal recovery again to the first $n-2t$ positions of the encoding of $|a\rangle$, we extract $|a\rangle$ and leave a corrupted encoding of $|0\rangle$:

$$|a\rangle \otimes \Big((E_{j_1}\mathcal{E}|0\rangle) \otimes \cdots \otimes \left(E_{j_{n-2t}}\mathcal{E}|0\rangle\right) \ \otimes \ E_{j_{n-2t+1}}^{(n-2t+1)} \cdots E_{j_n}^{(n)} \left(\mathbb{I}^{\otimes(n-2t)} \mathcal{E}^{\otimes 2t}\right) E_{j_0}\mathcal{E}|0\rangle \Big)$$

Effect of real reconstruction Now consider the effect of the decoding operator, which must be applied without knowledge of the positions which are corrupted. The first operation to be performed is to attempt to decode each branch $i \notin B$. This means copying the syndrome j_i for each branch into an ancilla state $|j_i\rangle$. Whenever E_{j_i} acts on a set $\tilde{B}_i$ such that $|\tilde{B}_i \cup B_i| \leq t$, then E_{j_i} can be identified and corrected. When E_{j_i} acts on too many positions, then it cannot be identified uniquely and the decoding procedure will simply leave that branch untouched.

Let I be the set of indices not in B which had few enough errors to correct. At the end of this first phase the input basis state will become:

$$\left((\prod_{i \notin I} E_{j_i}^{(i)}) \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E} |a\rangle \right) \otimes \bigotimes_{i \in I} |j_i\rangle$$

We know that all the honest players not in B are in I (by assumption of 2-GOOD-ness) and so I contains at least $n-2t$ positions. Decoding each of these circuits and applying the interpolation operator to the resulting qupits, we can extract the state $|a\rangle$ and replace it with $|0\rangle$ in the top-level sharing. This yields

$$|a\rangle \otimes \left((\prod_{i \notin I} E_{j_i}^{(i)}) \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E} |0\rangle \right) \otimes \bigotimes_{i \in I} |j_i\rangle$$

In both cases, the output can be written as $|a\rangle$ tensored with some ancilla whose state depends only on the syndromes $j_0, j_1, ..., j_n$. Once that state is traced out, the outputs of both operators will be identical. Another way to see this is that the ideal operator can simulate the real operator: one can go from the output of the ideal operator to that of the real operator by applying a transformation only to the ancilla. □

D.3.3 Simulatability

The previous two sections prove that the protocol satisfies an intuitive definition of security, namely that it is complete, sound and private. In this section, we sketch a proof that the protocol satisfies a more formal notion: it is equivalent to a simple ideal model protocol. The equivalence is statistical (Definition 3), that is the outputs of the real and ideal protocols may not be identical, but have very high fidelity to one another.

An Ideal Model Protocol Now, it is fairly simple to give an ideal protocol for VQSS: in the sharing phase, the dealer D sends his system S to $\mathcal{TTP}$. If D does not cooperate or sends an invalid message, $\mathcal{TTP}$ broadcasts "D is a cheater" to all players. In the reconstruction phase, $\mathcal{TTP}$ sends the system S to the designated receiver R. This protocol is in fact given in Protocol A.2 (p. 18).

Intuitively, this is the most we could ask from a secret sharing protocol: that it faithfully simulates a lock box into which the dealer drops the system he wishes to share.

In order to show equivalence of our protocol to the ideal protocol, we will give a transformation that takes an adversary $\mathcal{A}_1$ for our protocol and turns it into an adversary $\mathcal{A}_2$ for the ideal protocol. To give the transformation, we exhibit a simulator $\mathcal{S}$ which acts as an intermediary between $\mathcal{A}_1$ and the ideal protocol, making $\mathcal{A}_1$ believe that it is in fact interacting with the real protocol.

We give a sketch of the simulation procedure in Algorithm 1. Why does this simulation work?

- When D is cheating:

Algorithm 1. Simulation for VQSS (Protocol 3.1)

Sharing/Verification phase

- If D is a cheater, $\mathcal{S}$ must extract some system to send to $\mathcal{TTP}$:
  1. Run Sharing and Verification phases of Protocol 3.1, simulating honest players. If D is caught cheating, send "I am cheating" from D to $\mathcal{TTP}$.
  2. Choose $n-2t$ honest players not in B and apply ideal interpolation circuit to extract a system S.
  3. Send S to $\mathcal{TTP}$.
- If D is honest, $\mathcal{S}$ does not need to send anything to $\mathcal{TTP}$, but must still simulate the sharing protocol.
  1. Simulate an execution of the Sharing and Verification phases of Protocol 3.1, using $|0\rangle$ as the input for the simulated dealer D'.
  2. Choose $n-2t$ honest players (they will automatically not be in B since they are honest) and apply the ideal interpolation circuit to extract the state $|0\rangle$.
  3. The honest D will send a system S to $\mathcal{TTP}$.

Note: Regardless of whether D is honest or not, at the end of the sharing phase of the simulation, the joint state of the players' shares is a tree that is (essentially) 2-GOOD in both bases, and to which the ideal interpolation operator has been applied. Let I be the set of $n-2t$ honest players (not in B or C) who were used for interpolation.

Reconstruction phase

- If R is a cheater, $\mathcal{S}$ receives the system S from $\mathcal{TTP}$. He runs the interpolation circuit backwards on the positions in I, with S in the position of the secret. He sends the resulting shares to R.
- If R is honest, the cheaters send their corrupted shares to $\mathcal{S}$. These are discarded by $\mathcal{S}$.

In both cases, $\mathcal{S}$ outputs the final state of $\mathcal{A}_1$ as the adversary's final state.

  - When R is cheating, the simulation is trivially faithful, since there is *no difference* between the simulation and the real protocol: $\mathcal{S}$ runs the normal sharing protocol, then runs the interpolation circuit, sending the result to TTP. In the reconstruction phase, $\mathcal{S}$ gets the same state back from TTP, and runs the interpolation circuit backwards. Thus, the two executions of the interpolation circuit cancel out.
  - When R is honest, the faithfulness of the simulation comes from Lemma 3.2: in the real protocol, R outputs the result of the regular decoding operator. In the simulation, R gets the output of the ideal interpolation. Since the shared state has high fidelity to a 2-GOOD tree (by Lemma 3.1), the outputs will be essentially identical in both settings (i.e. they will have high fidelity).

- When D is honest:
  - To see that the simulation works when D is honest, we must show that two versions of the protocol are equivalent: in the first version, $\mathcal{S}$ gets S *after* having simulated the sharing phase with $\mathcal{A}_1$, and so he "swaps" it in by first running the ideal interpolation circuit, exchanging the system S for the shared state $|0\rangle$, and then running the interpolation circuit backwards.

    In the second version, he gets the system S from $\mathcal{TTP}$ *before* running the simulated sharing phase, and so he simply runs it with S as the input for the simulated dealer D'.

To see that the two versions are equivalent, view the "swap" as an atomic operation, i.e. view the application of the interpolation, switching out the $|0\rangle$ state and replacing it with S, and reapplying the interpolation backwards, as a single step. Now consider moving the swap backwards through the steps of the protocol. Because each of the verification steps acts as the identity on the shares of the honest players, we can move the swap backwards through all verifications (Note: the verification acts as the identity only when the dealer is honest, but that is indeed the case here). Finally, one can see by inspection that sharing a $|0\rangle$ and then swapping is the same as sharing the system S. Thus the two versions of the protocol are equivalent, and so the simulation is faithful when D is honest.

We have essentially proved:

Theorem D.9. *Protocol 3.1 is a statistically secure implementation of verifiable quantum secret sharing (Protocol A.2).*

D.4 Simulation for MPQC

Proof (of Theorem 5.4): The proof of this is by simulation, as before. The key observation is that when the simulator $\mathcal{S}$ is controlling the honest players, the adversary cannot tell the difference between the regular protocol and the following ideal-model simulation:

1. $\mathcal{S}$ runs the input phase as in the protocol, using $|0\rangle$ as the inputs for honest players. In this phase, if any dealer is caught cheating, $\mathcal{S}$ sends "I am cheating" to the $\mathcal{TTP}$ on behalf of that player.
2. $\mathcal{S}$ "swaps" the cheaters' inputs with bogus data $|0\rangle$, and sends the data to the $\mathcal{TTP}$. That is, he applies the interpolation circuit to honest players' shares to get the various input systems S_i (for $i \in \mathcal{C}$), and then runs the interpolation circuit backwards, with the state $|0\rangle$ replacing the original data.
3. $\mathcal{S}$ now runs the computation protocol with the adversary on the bogus data. (Because no information is revealed on the data, the adversary cannot tell this from the real protocol.)
4. $\mathcal{S}$ receives the true computation results destined to cheating players from $\mathcal{TTP}$.
5. $\mathcal{S}$ "swaps" these back into the appropriate sharings, and sends all shares of the i^{th} wire to player i (again, he does this only for $i \in \mathcal{C}$).

The proof that this simulation succeeds follows straightforwardly from the security of the top-level sharing protocol and the previous discussion on fault-tolerant procedures. □