Multi-party Quantum Computation

by

Adam Smith

B.Sc. Mathematics and Computer Science
McGill University, 1999.

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Author . . . .
Department of Electrical Engineering and Computer Science
August 22, 2001

Certified by . . . .
Madhu Sudan
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Multi-party Quantum Computation

by

Adam Smith

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

We investigate definitions of and protocols for multi-party quantum computing in the scenario where the secret data are quantum systems. We work in the quantum information-theoretic model, where 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 establish that any multi-party quantum computation can be securely performed as long as the number of dishonest players is less than $n/6$.

This thesis is based on joint work with Claude Crépeau and Daniel Gottesman.

Thesis Supervisor: Madhu Sudan
Title: Associate Professor

Acknowledgements

Thanks to my co-authors, Claude Crépeau and Daniel Gottesman, for their invaluable help with this research. Thanks to my eternally patient supervisor, Madhu Sudan. Thanks also to the many students, family mmbers, faculty and friends who provided encouragement, support and advice.

Contents

1 Introduction 11

1.1 Previous Work 14

1.2 Definitions 18

1.2.1 “Real” Model for Protocols 18

1.2.2 “Ideal” Model For Protocols 19

1.2.3 Protocol Equivalence 19

1.2.4 Static versus Adaptive Adversaries 20

1.2.5 Multi-party Quantum Computation 21

1.2.6 Verifiable Quantum Secret Sharing 21

1.3 Mathematical Preliminaries 23

1.3.1 Quantum Error-Correction 24

1.3.2 Sharing Quantum Secrets and (No) Cloning 27

1.3.3 Tools from Fault-Tolerant Quantum Computing 28

1.4 Neighborhoods of Quantum Codes 31

1.4.1 Well-Definedness of Decoding for States in $\mathcal{C}_B$ 33

2 Distributed Protocols for Quantum Computers 35

2.1 Subspace Projection 35

2.1.1 Completeness 36

2.1.2 Soundness 37

2.1.3 Dual Subspace Projection 39

2.2 VQSS Protocol: Two-Level Quantum Sharing 40

2.2.1 Sharing Shares: 2-GOOD Trees 40

2.2.2 Classical VSS 42

2.2.3 VQSS Protocol 42

2.2.4 (Informal) Soundness 45

2.2.5 (Informal) Completeness 49

2.2.6 Simulatability 49

2.2.7 Round and Communication Complexity 52

2.2.8 Additional Properties of Two-Level Sharing 53

2.3 Impossibility of VQSS when $t \geq \frac{n}{4}$ 54

2.4 Multi-party Quantum Computation 55

2.4.1 Level 3 Sharing Protocol 55

2.4.2 Distributed Computation 57

3 Open Questions **61**

A More on Neighborhoods of Quantum Codes **63**

List of Figures

1-1 Protocol 1 (Multi-party Quantum Computation—Ideal Model) 21
1-2 Protocol 2 (Verifiable Quantum Secret Sharing—Ideal Model) 22

2-1 Protocol 3 (Subspace Projection) 36
2-2 Algorithm 1 (Reconstruction for a 2-GOOD tree) 41
2-3 Protocol 4 (Modified VSS protocol from [CCD88]) 43
2-4 Protocol 5 (VQSS—Sharing Phase) 44
2-5 Protocol 6 (VQSS—Reconstruction Phase) 45
2-6 Algorithm 2 (Simulation for VQSS) 51
2-7 Protocol 7 (Top-Level Sharing) 56
2-8 Protocol 8 (Ideal Secret Sharing) 58
2-9 Protocol 9 (Multi-party Quantum Computation) 59

Chapter 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. When we approach distributed cryptography from the perspective of quantum computing, a number of natural questions arise:

- Do existing classical protocols remain secure when the adversary has access to a quantum computer?
- Can we use quantum computing and communication to find new, more secure or faster protocols for classical tasks?
- What new, *quantum* cryptographic tasks can we perform?

This research is inspired by the last of these questions. We propose to investigate a quantum version of an extensively studied classical problem, *secure multi-party computation* (or *secure function evaluation*), first introduced by [GMW87]. In this scenario, there are n players in a network. Each player i has an input x_i, and the players want to run a protocol to collectively compute some joint function $f(x_1, ..., x_n)$. The challenge is that all players would like this function evaluation to be *secure.* Informally, this means:

1. *Soundness and Completeness:* At the end of the protocol, all honest players should learn the correct function value $f(x_1, ..., x_n)$.
2. *Privacy:* Cheating players should learn nothing at all beyond what they can deduce from the function output and their own inputs.

Multi-party Quantum Computation For this thesis, 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 in n components $\mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_n$ such that P_i receives the output $\mathcal{H}_i$. Some components $\mathcal{H}_i$ may be empty.

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[1]. 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 as before. On one hand, 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. On the other hand, no coalition of t or fewer cheaters should be able to learn anything beyond what they can deduce from their initial knowledge of their input and from the systems $\mathcal{H}_i$ to which they have access. We formalize this notion in Section 1.2.

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 [CGMA85] 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. When the dealer passes the first phase of the protocol, then

- *Soundness:* There is a uniquely defined value s which will be reconstructed in the second phase, regardless of any interventions by an adversary who can control no more than t players.
- *Completeness:* If the dealer is honest, then he always passes the commitment phase and the value s recovered in the second phase is the secret he intended to share.
- *Privacy:* If the dealer is honest, no coalition of t players can learn any information about s.

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 a coalition of 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.

[1] For quantum information, merely having a state in one's possession in not the same as knowing a description of it, since one cannot completely measure an unknown quantum state

- *Privacy:* When D is honest, the adversaries learn no information about his input until the recovery phase.

Note that the privacy condition in this informal definition is redundant, by the properties of quantum information: any information adversaries could obtain about the shared state would imply some kind of disturbance (in general) of the shared state, which would contradict the completeness requirement. A formal definition of security is given in Section 1.2.

Contributions The results of this thesis are based on unpublished joint work with Claude Crépeau and Daniel Gottesman [CGS01]. In this thesis:

- 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.

Our protocols run in time polynomial in both n, the number of players, and k, the security parameter. The error of the protocols (to be defined later) is exponentially small in k.

Beyond these specific results, there are a number of conceptual contributions of this thesis 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 [GL90, Bea91, MR91, Can00, DM00, CDD$^+$01, PW00, Can01, vdG97].
- 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 [CCD88]. The key property these 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.

Organization The thesis is organized as follows. Chapter 1 contains the material necessary for understanding the protocols of this thesis as well as their context. Section 1.1 describes the previous work on the topics in this thesis, with emphasis on the works whose results we use directly. In Section 1.2, we present a framework for defining security of a distributed quantum protocol which involves interaction with a trusted third party. We use this framework to formally define both verifiable quantum secret sharing and multi-party quantum computation. Section 1.3 contains the mathematical background for understanding our protocols, as well as results we use from the existing literature. In Section 1.4, we introduce three definitions of the local "neighborhoods" of a quantum code, in order to help the reader understand exactly what properties our protocols guarantee and what properties are needed in our security analyses. Some additional relations between these three notions are shown in Appendix A.

The protocols which are the main focus of this thesis are presented in Chapter 2. One of the main proof techniques we use is a "quantum-to-classical reduction" (terminology due to [LC99]). In Section 2.1, we illustrate this technique with a simple protocol which achieves VQSS for a small number of cheaters ($t < n/8$), and whose analysis will prove insightful for the sequel. Section 2.2 uses a similar technique, but applied to a modified version of the classical "verifiable blob" protocol of [CCD88], to construct a VQSS protocol secure against $t < n/4$ cheaters. In Section 2.3, we show this is optimal by relating VQSS protocols to error-correcting codes and applying the quantum Singleton bound. Finally, we use our sharing scheme to contruct MPQC protocol which tolerates any $t < n/6$ cheaters.

We conclude with some open questions related to our results (Chapter 3).

1.1 Previous Work

Classical MPC Most of the work on classical distributed protocols is based on *secret sharing*, in which a message is encoded and shared amongst a group of players such that no coalition of t players gets any information at all about the encoded secret, but any group of $t + 1$ or more players can recover the secret exactly. The prototypical and most commonly used solution to this is the polynomial sharing scheme due to Shamir [Sha79]: choose a random polynomial p of degree at most t over $\mathbb{Z}_p$ (for some prime $p > n$) subject to $p(0) = s$, where s is the secret being shared. The share given to player i is value $p(i)$, for $i = 1, ..., n$. Note that for normal secret sharing we assume that the shares are prepared honestly.

This assumption was removed in subsequent work: Multi-party computing, in which no player may be assumed to be honest, was first treated explicitly by Goldreich et al. [GMW87], although the subtask of verifiable secret sharing had been investigated previously by Chor et al. [CGMA85]. Goldreich et al. [GMW87] proved that *under computational assumptions*, secure multi-party evaluation of any function was possible tolerating any minority of cheating players, i.e. for any $t < \frac{n}{2}$.

Subsequently, Ben-Or et al. [BGW88] and Chaum et al. [CCD88] independently proved that tolerating up to $t < \frac{n}{3}$ was possible *without* computational assumptions, provided that one assumed that every pair of participants was connected by a secure channel. Moreover, this bound is tight due to the impossibility of even agreeing on a single bit when $t \geq \frac{n}{3}$ (see Lynch [Lyn96], for example). The main difference between the results of [CCD88] and those of [BGW88] is that the former allow a small probability of error (exponentially small in the complexity of the protocol).

The bound of $\frac{n}{3}$ for information-theoretically secure MPC was broken by Rabin and Ben-Or [RB89] and Beaver [Bea89], who showed that assuming the existence of a secure broadcast channel, then one can in fact tolerate any minority ($t < \frac{n}{2}$) of cheaters without computational assumptions. Their protocols introduce a small error probability, which is provably unavoidable [RB89]. The results of [RB89, Bea89] were extended to the model of adaptive adversaries by Cramer et al. [CDD$^+$99].

All of these protocols rely on verifiable secret sharing. Our solution draws most heavily on the techniques of [CCD88]. The essential idea behind their VSS protocol is to share the secret using a two-level version of the basic scheme of Shamir (above), and then use a cut-and-choose zero-knowledge proof to allow the dealer to convince all players that the shares he distributed were consistent with a single polynomial $p(x)$.

Beyond these basic protocols, a line of work has focused on coming up with proper definitions of multi-party computing [GL90, Bea91, MR91, Can00, DM00, CDD$^+$01, PW00, Can01]. Both [Can01] and [CDD$^+$01] provide summaries of that literature. Most of the research has focused on finding definitions which allow composability of protocols, mainly focusing on multi-party computing (often referred to, more precisely, as *secure function evaluation*). In this work, we adopt a simple definition (based on the initial definitions of Canetti). We do not prove any composition protocols, but simply ensure that the definition captures our intuition of security and is provably achieved by our protocols. See Section 1.2 for further discussion.

Multi-party Quantum Protocols Relatively little work exists on multi-party cryptographic protocols for quantum computers. Secret sharing with a quantum secret was first studied by Cleve, Gottesman and Lo [CGL99]. They suggested a generalization of the Shamir scheme, which is also used by Aharonov and Ben-Or [AB99] as an error-correcting code. One of the contributions of [CGL99] was that to point out the strong connection between secret sharing and error-correcting codes in the quantum setting (see Section 1.3.2). Our VQSS protocol is based on the [CGL99] scheme, using a modification of the techniques of [CCD88] to ensure the consistency of distributed shares.

There were some additional works on distributed quantum protocols. Gottesman [Got00] showed that quantum states could be used to share classical secrets more efficiently than is possible in a classical scheme. Chau [Cha00] proposed a scheme for speeding up *classical* multi-party computing using quantum techniques; [Cha00] also mentions the problem of verifiable quantum secret sharing as an open question. The dissertation of van de Graaf [vdG97] discusses defining the security of classical distributed protocols with respect to a quantum adversaries, but contains no constructions.

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.

One can view multi-party computation as fault-tolerant computing with a different error model, one that is suited to distributed computing. On one hand, 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.

On the other hand, the error model of MPQC is stronger in some respects: 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 in a model of adversarially placed errors.

Techniques from FTQC are nonetheless useful for multi-party computing. Considerable research has been done on FTQC. We rely mainly on the techniques of Aharonov and Ben-Or [AB99], which were based on those of Shor [Sho96]. Using "CSS" quantum error-correcting codes, Shor showed that fault-tolerance was possible so long as the error rate in the computer decreased logarithmically with the size of the computation being performed. Aharonov and Ben-Or showed that by using concatenated coding, one could in fact tolerate a constant error rate. They also introduced generalized CSS codes in which the individual pieces of a codeword are assumed to be higher-dimensional systems, such as collections of several qubits (this corresponds to using larger alphabets in classical coding theory).

Provably Secure (and Insecure) Quantum Protocols While quantum cryptographic protocols have existed for some time, many of them have been proven secure only recently. The first proofs of security appeared in the context of entanglement purification protocols [BBP$^+$96, DEJ$^+$96, LC99]. In a different line of work, Mayers [May98] provided a notoriously difficult proof that the Bennett-Brassard key distribution scheme was secure. Unifying these two lines of research, Shor and Preskill [SP00] proved the correctness of the Bennett-Brassard [BB84] key distribution protocol, based on a previous proof of a purification-based protocol due to Lo and Chau [LC99]. The main insight of [LC99] was that in certain situations, proving the security of a quantum protocol could be reduced to classical probability arguments, since one could assume without loss of generality that the adversary followed one of a finite number of classical cheating strategies (a so-called "quantum-to-classical reduction").

A similar technique is used in [BCG$^+$01] to prove the correctness of a scheme for authenticating quantum transmission. This technique will also be useful for proving the soundness of our protocol, as it will allow us deal with possible entanglement between data and errors by "reducing" them to classical correlations.

Note that for protocols where the adversary is one of the participants in the system and not an outside eavesdropper, much less is known. Some proofs were also attempted for tasks such as bit commitment [BCJL93], but those proofs were later discovered to be flawed, since bit commitment was proven impossible [May96, LC97a, May97, LC96, LC97b, BCMS98]. There have also been several works on quanutm coin-tossing. Although arbitrarily small error is known to be impossible, several works have focused on reducing the error as much as possible [LC96, MS99, ATVY00, Amb01]. Yet another line of work has focused on how to achieve certain two-party tasks using computional assumptions, i.e. assuming that there exist (quantum) one-way permutations [DMS00, CLS01].

1.2 Definitions

This section describes a simple framework for proving the security of distributed quantum cryptographic protocols. The defintions are based on the initial framework of Canetti [Can00], as well as on discussions in the dissertation of van de Graaf [vdG97]. 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 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.

1.2.1 "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 [BGW88, CCD88][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, 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

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

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").

1.2.2 "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 $Ideal(\mathcal{A}, \rho)$.

1.2.3 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 1 (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 2 (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 [vdG97]: $\mathcal{S}$ 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 2.2.6 and Section 2.4.2.

1.2.4 Static versus Adaptive Adversaries

In this thesis, 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 [CFGN96, CDD$^+$99].

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 2.1.2, we discuss briefly how some of the proofs could be extended to handle adaptivity (see Remark 4, p. 38).

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

1.2.5 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 (see Open Questions, Chapter 3).

Protocol 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 $\mathcal{TTP}$.

Figure 1-1: Protocol 1 (Multi-party Quantum Computation—Ideal Model)

1.2.6 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.

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 thesis, 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 Figure 1-2. 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.

Protocol 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.

Figure 1-2: Protocol 2 (Verifiable Quantum Secret Sharing—Ideal Model)

1.3 Mathematical Preliminaries

We assume that the reader is familiar with the basic notation and formalism of quantum computing. For an introduction, the reader should refer to a textbook such as Nielsen and Chuang [NC00].

For most of this paper, we will work with "qupits", that is p-dimensional quantum systems, for some prime p. It is natural to view the elements of the field $F = \mathbb{Z}_p$ as a basis for the state space of a qupit.

In our settings, it will be useful to choose p so that $n < p$. We need not choose p very big for this, since there is always a prime between n and $2n$. However, all of our protocols will remain polynomial time even when p is exponential in n. That is, the complexity of the protocols will be polynomial in $\log |F| = \log p$.

Just as for the case of qubits, there are a few natural operators on qupits which we will use extensively in this paper.

The shift and phase operators for qupits (sometimes denoted σ_x, σ_z) are defined analogously to the case of qubits:

$$X|a\rangle \mapsto |a+1 \mod p\rangle \quad \text{and} \quad Z|a\rangle \mapsto \omega^a|a\rangle,$$

where $\omega = e^{2\pi i/p}$. These two operators generate the Pauli group. Since they have a simple commutation relation ($XZ = \omega ZX$), any element of the group is proportional to some product $X^x Z^z$ for $x, z \in \{0, ..., p-1\}$. As for qubits, the p^2 operators $X^x Z^z$ form a basis for the space of $p \times p$ complex matrices, and so any unitary operator on qupits can be written as a linear combination of Pauli matrices. In particular, this is useful since means that correcting Pauli errors in a quantum code is sufficient for correcting arbitrary errors. In the context of errors, X is called a *shift error* and Z is a *phase error*.

For registers of qupits, the Pauli matrices are tensor products of Pauli matrices acting on individual qupits. If $\mathbf{x} = (x_1, ..., x_n)$ and $\mathbf{y} = (y_1, ..., y_n)$ are vectors in $\mathbb{Z}_p^n$, then $X^{\mathbf{x}} Z^{\mathbf{z}}$ denotes $X^{x_1} Z^{z_1} \otimes \cdots \otimes X^{x_n} Z^{z_n}$. These form a basis for the space of operators on the register. The set of positions on which a Pauli matrix does *not* act as the identity is called its *support*, and is equal to the union of the supports of $\mathbf{x}$ and $\mathbf{z}$. The number of such positions is called the *weight* of the operator.

Fourier Rotations Another transformation which arises often is the Fourier transform on qupits, which generalizes the Hadamard rotation on qubits.

$$\mathcal{F}|a\rangle \mapsto \sum_{b \in \mathbb{Z}_p} \omega^{ab}|b\rangle$$

This is called a Fourier rotation since its effect on the p-dimensional vector of coefficients of the state of a qupit is exactly that of the Fourier transform over the group $\mathbb{Z}_p$. Consequently, phase changes become shifts in this new basis, and conversely:

$$\mathcal{F}X = Z\mathcal{F} \quad \text{and} \quad \mathcal{F}Z = X^{-1}\mathcal{F}$$

A useful property of the Fourier transform is that linear transformations remain linear after the change of basis. Specifically, let V be an invertible $n \times n$ matrix over $\mathbb{Z}_p$. Let V denote the corresponding unitary operator on a register of n qupits, i.e $\widetilde{V}|\mathbf{x}\rangle = |\mathbf{Vx}\rangle$. Then in the Fourier basis, this looks like a different linear map, given by the matrix $(V^{-1})^\top$. That is $\mathcal{F}\widetilde{V}\mathcal{F}^{-1} = (\widetilde{V^{-1})^\top})$.

The main feature we will use is simply that the transformation remains a linear permutation of the basis vectors. There is one very useful special case. For controlled addition (denoted c-X), which maps $|a, b\rangle \mapsto |a, a+b\rangle$, conjugating by a Fourier rotation yields another controlled-addition, applied in the opposite direction and with a scaling factor of -1 (i.e. $|a, b\rangle \mapsto |a-b, b\rangle$).

1.3.1 Quantum Error-Correction

A quantum error-correcting code is a way of encoding redundancy into quantum information to allow correction of errors which occur during transmission or storage. An $[[n, k, d]]$ quantum code encodes k qubits into n qubits (for $n \geq k$) and corrects any (arbitrary) error which affects less than $\frac{d}{2}$ positions in the code. The most resilient quantum codes actually work over higher-dimensional subspaces, i.e. each "position" in the code consists of a qupit. Recall that we work with qupits of dimension p, where p is some prime greater than n.

Css Codes An important family of quantum codes are the css codes (due to Calderbank-Shor [CS96] and Steane [Ste96]). A css code over n qupits is defined by two classical linear codes V and W over $\mathbb{Z}_p$, both of length n. They are chosen such that $V^\perp \subseteq W$, where $V^\perp$ is the dual of V with respect to the standard dot product $v \cdot w = \sum_{i=1}^{n} v_i w_i$. Note that we automatically also have $W^\perp \subseteq V$. The quantum code $\mathcal{C}$ is then the set of states $|\psi\rangle$ which would yield a codeword of V if they were measured in the computational basis ($\{|0\rangle, |1\rangle, ..., |p-1\rangle\}$), and yield a codeword of W if they were measured in the Fourier basis ($\{\mathcal{F}|0\rangle, ..., \mathcal{F}|p-1\rangle\}$).

Now for any given system of n qupits and any linear subspace $W \leq F^n$, we define

$$W^{(q)} = \text{span}\{|\mathbf{w}\rangle : \ \mathbf{w} \in W\}.$$

If we denote by $\mathcal{F}^{\otimes n}$ the parallel application of $\mathcal{F}$ to all qubits of an n-qubit register, then we have:

$$\mathcal{C} = V^{(q)} \cap \mathcal{F}W^{(q)}$$

The dimension of $\mathcal{C}$ as a code, i.e. number of qupits it can encode, is simply $\dim(V/W^\perp) = \dim V - \dim W^\perp$. For convenience, we will denote $V_0 = W^\perp$ and $W_0 = V^\perp$, and so the formula for the number of qupits encoded becomes $\dim V - \dim V_0 = \dim W - \dim W_0$.

Minimum Distance To correct an arbitrary error on a subset A of positions ($A \subseteq \{1, ..., n\}$), it turns out that it is sufficient (and necessary) to be able to correct Pauli errors, i.e. compositions of shift and phase errors applied to the qupits in A. Thus, to correct errors on any t positions it suffices to correct all Pauli errors of weight at most t. A sufficient condition is that the spaces $\{E\mathcal{C}\}$ be mutually orthogonal, where E ranges over all Pauli operators of weight at most t. In such a case, one can correct any of these errors E on a codeword $|\psi\rangle$ by performing a measurement that identifies which of these subspaces contains the corrupted codeword $E|\psi\rangle$, and then applying the correction E^{-1}. This can be rephrased: for all Pauli operators of weight at most $2t$, $E\mathcal{C}$ and $\mathcal{C}$ should be orthogonal spaces. The minimum distance of a quantum code C is thus the weight of the smallest Pauli operator for which this is not true.

Definition 3. *The minimum distance of a quantum code $\mathcal{C}$ is the weight of the smallest Pauli operator such that $\mathcal{C}$ and $E\mathcal{C}$ are not orthogonal.*

By the previous discussion, a code with distance d can correct arbitrary errors on any $\lfloor (d-1)/2 \rfloor$ positions. For CSS codes, there is a simple way to calculate the minimum distance:

Fact 1.1. *Let V, W be classical codes with minimum distances d_1 and d_2 such that $V^{\perp} \subseteq W$. Then the quantum* CSS *code $\mathcal{C} = V^{(q)} \cap \mathcal{F}W^{(q)}$ has minimum distance at least* $\min(d_1, d_2)$. [4]

Syndromes and Error Correction Given a classical linear code V of dimension k, the syndrome for V is a linear function from n bits to $n-k$ bits that indicates which coset of V contains its argument. If V has distance at least $2t+1$ and a codeword $\mathbf{v} \in V$ is altered in t or fewer positions, then the syndrome of the corrupted word $\mathbf{v} + \mathbf{e}$ allows one to compute the correction vector $-\mathbf{e}$. We will let V-syndrome denote the syndrome with respect to V. Note that computing the V-syndrome is easy. Fix a basis $\{\mathbf{v}_1, ..., \mathbf{v}_{n-k}\}$ of the dual code $V^{\perp}$. The V-syndrome of $\mathbf{w}$ is the vector $(\mathbf{v_1} \cdot \mathbf{w}, ..., \mathbf{v}_{n-k} \cdot \mathbf{w})$.

This is the basis for the error correction procedure for CSS codes. Suppose that $E = X^{\mathbf{x}} Z^{\mathbf{z}}$, and both $\mathbf{x}$ and $\mathbf{z}$ have support on at most t positions. Let $|\psi\rangle \in \mathcal{C}$. Since $|\psi\rangle$ lies in $V^{(q)}$, measuring the V-syndrome of $E|\psi\rangle$ (in the computational basis) allows one to compute the vector $\mathbf{x}$, and apply the correction $X^{-\mathbf{x}}$. Similarly, measuring the W-syndrome in the Fourier basis allows one to compute $\mathbf{z}$ and apply $Z^{-\mathbf{z}}$, thus recovering $|\psi\rangle$. The two measurements commute, so in fact it does not really matter which one is applied first.

The pair of measurement results used, namely the V-syndrome in the computational basis and the W-syndrome in the Fourier basis, are referred to together as the *quantum syndrome.* If the syndromes are s_1 and s_2 pits long respectively, then there are $p^{s_1+s_2}$ possible quantum syndromes. This divides the whole space $\mathbb{C}^{\mathbb{Z}_p^n}$ into $p^{s_1+s_2}$ orthogonal subspaces indexed by the set of *equivalence classes* of Pauli operators. That is, two Pauli operators E, E' are deemed equivalent if $E\mathcal{C} = E'\mathcal{C}$; and the space $\mathbb{C}^{\mathbb{Z}_p^n}$ can be written as the direct sum of the orthogonal spaces $\{E_j\mathcal{C}\}_{j\in J}$, where J is

[4] *In fact, the minimum distance of $\mathcal{C}$ is the minimum of the weights of the lightest vectors in $V - V_0$ and $W - W_0$. These are bounded below by the minimum distances d_1, d_2, and the bound is tight for the codes used in this paper.*

a set of indices which contains exactly one element from each equivalence class. For a CSS code, two Pauli operators $X^{\mathbf{x}}Z^{\mathbf{z}}$ and $X^{\mathbf{x}'}Z^{\mathbf{z}'}$ will be equivalent if and only if $\mathbf{x}$ and $\mathbf{x}'$ are in the same coset of V, and $\mathbf{z}$ and $\mathbf{z}'$ are in the same coset of W.

Note that the dimension of the code can also be written as $n - s_1 - s_2$.

Quantum Reed-Solomon Codes In this work we will use a family of CSS codes known as "quantum polynomial codes" or "quantum Reed-Solomon codes". These were introduced by Aharonov and Ben-Or [AB99], and generalize classical Reed-Solomon codes.

In this paper, we will specify a quantum RS code by a single parameter $\delta < (n-1)/2$, which represents the degree of the polynomials used in the code. The corresponding code $\mathcal{C}$ will encode a single qupit and correct $t = \lfloor \frac{\delta}{2} \rfloor$ errors. For simplicity, choose $\delta = 2t$. We will always choose the number n of players to be either $2\delta + 1$ or $3\delta + 1$.

If n is the number of players, choose any p such that $p > n$ ([5]). We will work over the field $F = \mathbb{Z}_p$. The classical Reed-Solomon code V^δ is obtained by taking the vectors

$$\hat{\mathbf{q}} = (q(1), q(2), \dots, q(n))$$

for all univariate polynomials q of degree at most δ. The related code V_0^δ is the subset of V^δ corresponding to polynomials which interpolate to 0 at the point 0. That is:

$$\begin{array}{rcl} V^\delta & = & \{\hat{\mathbf{q}} : \ q \in F[x] : \ \deg(q) \leq \delta\} \\ V_0^\delta & = & \{\hat{\mathbf{q}} : \ \deg(q) \leq \delta \text{ and } q(0) = 0\} \subseteq V^\delta \end{array}$$

The code V^δ has minimum distance $d = n - \delta$. Moreover, errors (up to $\lfloor (n - \delta - 1)/2 \rfloor$ of them) can be corrected *efficiently*, given the syndrome of the corrupted word.

Note that by the non-singularity of the Vandermonde matrix (i.e. polynomial interpolation), there exists a vector $\mathbf{d} = (d_1, \dots, d_n) \in F^n$ such that $\mathbf{d}^\top \hat{\mathbf{f}} = f(0)$ for any $f \in F[x]$ and $deg(f) < n$.

Fact 1.2. *Let $\delta' = n - \delta - 1$. The duals of the codes V^δ, V_0^δ are*

$$\begin{array}{rcccl} W^{\delta'} & = & (V_0^\delta)^\perp & = & \{(d_1 q(1), ..., d_n q(n)) : \ \deg(q) \leq \delta'\} \\ W_0^{\delta'} & = & (V^\delta)^\perp & = & \{(d_1 q(1), ..., d_n q(n)) : \ \deg(q) \leq \delta' \textit{ and } q(0) = 0\} \end{array}$$

Thus the dual of a Reed-Solomon code of degree δ is another RS code with degree δ', but where each component has been "scaled" according to some constant d_i. One can also show that $d_i \neq 0$ for all i.

The code $\mathcal{C}$ for parameter δ (occasionally written $\mathcal{C}^\delta$) is the CSS code obtained from codes $V = V^\delta$ and $W = W^{\delta'}$. As mentioned before, it encodes a single qupit since $\dim V = \delta + 1$ and $\dim W^\perp = \delta$. Moreover, the minimum distance of V is $n - \delta$ and the minimum distance of W is $\delta + 1$. Thus, for $\delta < (n-1)/2$ we get that the minimum distance of $\mathcal{C}$ is at least $\delta + 1$, and it corrects at least $t = \delta/2$ errors.

[5] In fact, the construction can be changed to allow $p = n$.

The encoding we obtain can be described explicitly. Let $V_a^\delta = \{\hat{\mathbf{q}} : \deg(q) \leq \delta \text{ and } q(0) = a\}$. Then for any qupit in a pure state $|\psi\rangle = \sum_{a \in F} \alpha_a |a\rangle$, the encoded version is (ignoring normalization constants):

$$\mathcal{E}|\psi\rangle = \sum_a \alpha_a \mathcal{E}|a\rangle = \sum_a \alpha_a \sum_{\mathbf{v} \in V_a^\delta} |\mathbf{v}\rangle = \sum_a \alpha_a \sum_{q:\deg(q) \leq \delta,\ q(0)=a} |q(1), ..., q(n)\rangle$$

Note that the circuit for encoding is very simple: consider the linear map which takes the coefficients of a polynomial of degree at most δ and maps it to the vector $q(1), ..., q(n)$. Then placing $|\psi\rangle$ in the position of the constant coefficient and initializing all other coefficients to the equal superposition $\sum_a |a\rangle$ will yield the output $\mathcal{E}|\psi\rangle$.

Correction, Detection and Erasures As mentioned above, the classical RS codes have efficient decoding algorithms for identifying and decoding the maximum number of errors which is information-theoretically possible, i.e. t where $d = 2t + 1$ is the minimum distance. Consequently, so do the quantum polynomial codes, since for CSS codes one simply corrects errors in each of the two bases.

They can also *detect* up to $2t$ errors, at the expense of correction. Simply measure a received codeword to see if its syndrome is 0. If a non-zero Pauli operator of weight less than d has been applied to the word, the syndrome will be non-zero, and the error will be detected. For an arbitrary error of weight less than d, the projection of the corrupted word onto the code will be exactly the original codeword.

Remark 1. In some of our protocols, we will want to detect a large number of errors, but still be able to correct a small number. Suppose that we have identified b positions which are known to be corrupted (for example, say they have been erased). Then the quantum polynomial code will be able to identify t *further* errors, and will able to correct them if there are at most $t - b$.

(That is, the punctured code (i.e. restricted to the $n - b$ non-erased positions) has distance $2t + 1 - b$. Given a corrupted word, one can tell if it is within $t - b$ of a codeword, and correct such errors. If it is not within distance $t-b$ of a codeword, then more errors occurred. However, as long as less than t errors occurred, the corrupted word will not be within $t - b$ of anything but the correct codeword, since $t + (t - b)$ is less than the new minimum distance).

1.3.2 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.

Fact 1.3 (No cloning, folklore). *Let $U : \mathcal{H}_m \otimes \mathcal{H}_W \longrightarrow \mathcal{H}_A \otimes \mathcal{H}_B$ ([6]) 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 [CGL99]: 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 somehow "built in."

1.3.3 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 [Sho96, AB99, GC99].

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[7]. 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.

[6] *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.*

[7] 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.

For qupits of prime dimension p, Aharonov and Ben-Or [AB99] 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.

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 [Sho96, AB99].

Remark 2. 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 $n/3$, [AB99] 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 [Got] (based on [GC99]), and is much more efficient than the original one proposed in [AB99]. 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 2), 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).

1.4 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, so that error correction and detection can be used to correct any tampering, or at least detect it and identify its origin.

For classical codes, the notion of closeness is clear: the set of positions on which a real word $\mathbf{v}$ differs from a codeword provides a lot of information; in particular, the size of this set is the Hamming distance of $\mathbf{v}$ from the code. As long as the minimum distance of the code is at least $2t$, ensuring that $\mathbf{v}$ differs from a codeword only on the positions held by cheaters means that any errors introduced by cheaters will be correctable.

Given a set B of cheaters ($B \subseteq \{1, ..., n\}$), we define:

$$\begin{aligned} W_B &= \{\mathbf{v} \;:\; \exists \mathbf{w} \in W \text{ s.t. } \mathrm{supp}(\mathbf{v}-\mathbf{w}) \in B\} \\ &= \{\mathbf{v} \;:\; \exists \mathbf{w} \in W \text{ s.t. } \mathbf{v} \text{ differs from } \mathbf{w} \text{ only on positions in } B\} \end{aligned}$$

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) receiver/reconstructor.

Remark 3. *V_B is a linear code, and its dual is exactly the set of words in $V^\perp$ which have support included in the complement of C (say $A = \{1, ..., n\} \setminus B$). In particular, this means that if one wants to measure the V_B-syndrome, one only needs access to positions in A.*

For quantum codes, the situation is more complex. For a CSS code $\mathcal{C}$, there is more than one natural definition of the neighborhood corresponding to a set B of positions. Let $\mathcal{H} = \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$ be partitioned according to two sets A, B, so that $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$. We consider three definitions of an "B-neighborhood" of $\mathcal{C}$. Let ρ be an arbitrary state of $\mathcal{H}$.

For a mixed state given by density matrix ρ', we say ρ' is "in" $\mathcal{C}$ if all states in the mixture lie in $\mathcal{C}$ (no matter how the mixture is written). Algebraically, this is given by the condition $\mathrm{Tr}(P_{\mathcal{C}}\rho') = 1$ where $P_{\mathcal{C}}$ is the projector onto the subspace $\mathcal{C}$.[8]

1. ρ differs from a state in $\mathcal{C}$ only by some super-operator local to B:

$$N_B(\mathcal{C}) = \{\rho : \exists \rho' \text{ in } \mathcal{C}, \exists \mathcal{O} \text{ super-operator, acting only on } \mathcal{H}_B \text{ s.t. } \rho = \mathcal{O}(\rho')\}$$

2. ρ is cannot be distinguished from a state in $\mathcal{C}$ by looking only at positions in A. Algebraically, this is captured by requiring that the density matrix obtained by "tracing out" the positions in B be the same as for some state in the code (the notation ST stands for "same trace"):

$$ST_B(\mathcal{C}) = \{\rho : \exists \rho' \text{ in } \mathcal{C} \text{ s.t. } \mathrm{Tr}_B(\rho) = \mathrm{Tr}_B(\rho')\}$$

3. Specifically for CSS codes, one could simply 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, would yield the result 0. The set of states which pass this test is:

$$\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}.$$

These notions form a hierarchy, namely $N_B(\mathcal{C}) \subseteq ST_B(\mathcal{C}) \subseteq \mathcal{C}_B$. (The first inclusion holds since super-operators local to B do not change the density matrix of the components in A. The second inclusion holds since the outcome distribution of any tests local to A is determined entirely by $\mathrm{Tr}_B(\rho)$.) However, the three notions are distinct and in fact only one of them—notion (3)—always describes a linear subspace of $\mathcal{H}$. We discuss these three notions further in Appendix A.

In the analysis of quantum error-correction and fault-tolerance protocols, it is sufficient to consider notion (1). This stems from 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, the main

[8] To see why this is the case, write $\rho' = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ with $\langle\psi_i|\psi_j\rangle = \delta_{ij}$ and $\sum_i p_i = 1$. Then all the $|\psi_i\rangle$'s are in $\mathcal{C}$ if and only if $\langle\psi_i|P_{\mathcal{C}}|\psi_i\rangle = 1$. Taking the trace over the matrix $P_{\mathcal{C}}\rho$ yields 1 if and only if this condition holds.

contribution of this paper is the construction of protocols which guarantee conditions similar to (3) above. In Section 2.1, we illustrate the ideas with a simple protocol, dubbed *subspace projection*, which is sufficient for VQSS and MPQC when $t < n/8$. In Section 2.2, we give a VQSS protocol tolerating $t < n/4$, and we show that this tolerance is optimal in Section 2.3. Finally, in Section 2.4, we show how to ensure condition (3) above and how the techniques from fault-tolerant computing can then be used to achieve multi-party computation of an arbitrary quantum circuit when $t < n/6$.

1.4.1 Well-Definedness of Decoding for States in $\mathcal{C}_B$

In this section we prove a property of $\mathcal{C}_B$ which will be useful in the proof of security (and hopefully also provide some intuition for our construction).

Suppose that the minimum distance of $\mathcal{C}$ is $d > 2t+1$, and B is restricted in size: $|B| < t$. Then applying the usual decoding circuit for $\mathcal{C}$ without knowing exactly where B is yields the same result as applying an ideal interpolation circuit which first discards positions in B and then reconstructs the logical data as if it was handling a regular codeword. Formally, there are two natural "reconstruction operators" for extracting the secret out of a state which has been shared among several players.

1. $\mathcal{D}$ 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 $|j\rangle$.
2. $\mathcal{R}^I$ is the "ideal recovery operator", defined by identifying the set B of cheaters and applying the simple interpolation circuit to a set of $n-2t$ good players' positions.

Proposition 1.4. *For any state ρ in $\mathcal{C}_B$ where $|B| \leq t$, the state $\mathcal{R}^I(\rho)$ is well-defined and is equal to $\mathcal{D}(\rho)$.*

We give the proof of this below. For now, note that Proposition 1.4 means that no changes made only to the components in B—no matter how they might be made to interact with outside systems entangled with the data—will change the reconstructed state.

In order to prove Proposition 1.4, we characterize $\mathcal{C}_B$ algebraically:

Lemma 1.5. *Suppose that ρ has fidelity 1 to $\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}$. Then we can write*

$$\rho = \textstyle\sum_i p_i |\psi_i\rangle\langle\psi_i|$$
$$|\psi_i\rangle = \textstyle\sum_j c_{ij} E_j |\phi_{ij}\rangle$$

where E_j are Pauli operators on B and $|\phi_{ij}\rangle \in \mathcal{C}$.

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 1.5): 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 1.4): 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 1.5, the final outcome will be $\rho' = \sum_{ij} p_i |c_{ij}|^2 |\phi_{ij}\rangle\langle\phi_{ij}|$. □

Chapter 2

Distributed Protocols for Quantum Computers

2.1 Subspace Projection

Before presenting the main VQSS protocol, we describe 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 [CCD88] 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.

Recall that for any given system of n qupits and any linear subspace W of $F^n = \mathbb{Z}_p^n$, we define

$$W^{(q)} = \mathrm{span}\{|\mathbf{w}\rangle : \ \mathbf{w} \in W\}.$$

For this protocol, W can be any code with minimum distance $2t+1$ and an efficient decoding algorithm. However, 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 [1], 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 $W_{B \cup C}^{(q)}$.

We obtain a "cut-and-choose" protocol, also similar to the "random hashing"

[1] 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.

technique used in purification protocols (Protocol 3, Figure 2-1). Note that VSS and broadcast of classical data are not a problem since $t < \frac{n}{4} < \frac{n}{3}$ ([BGW88, CCD88, Lyn96]).

Protocol 3 (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 $\mathcal{H}_1, \dots, \mathcal{H}_k$ 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 $b_1, \dots, b_k$ (these are completely unpredictable to the dealer, even if he is cheating).
3. For $\ell = 1, \dots, k$, 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 $\mathcal{H}_1, \dots, \mathcal{H}_k$ in the computational basis and broadcast the result.
4. Each of the broadcasted words $\mathbf{w}_1, \dots, \mathbf{w}_k$ 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.

Figure 2-1: Protocol 3 (Subspace Projection)

2.1.1 Completeness

Lemma 2.1. *When the dealer D is honest, he will pass the protocol. Moreover, we will have $B \subseteq C$, i.e. only real cheaters will be accused of cheating.*

Proof: If the dealer is honest, he will use some $\mathcal{H}_0$ in $W^{(q)}$ and will have all $\mathcal{H}_\ell$'s in state $\sum_{\mathbf{w} \in W} |\mathbf{w}\rangle$. Consider some round ℓ. Now no matter what the value of b_ℓ is, applying $(c\text{-}X^{b_\ell})$ to *all* of the shares is equivalent to the identity on $\mathcal{H}_0 \otimes \mathcal{H}_\ell$, since for all $\mathbf{v} \in W$, we have:

$$(c\text{-}X^{b_\ell})|\mathbf{v}\rangle \sum_{\mathbf{w} \in W} |\mathbf{w}\rangle = |\mathbf{v}\rangle \sum_{\mathbf{w}} |\mathbf{w} + b_\ell \mathbf{v}\rangle = |\mathbf{v}\rangle \sum_{\mathbf{w}} |\mathbf{w}\rangle$$

Of course, in the protocol we can only guarantee that honest players will apply $(c_X^{b_\ell})$ to their shares of $\mathcal{H}_0$ and $\mathcal{H}_\ell$. Nonetheless, the result is the same as applying the identity to the honest players' shares. Consequently, the values broadcast at Step 3 by the honest players will all be consistent with some $\mathbf{w} \in W$. Since we've assumed that the distance of the code W is at least $2t + 1$, any false values broadcast by cheaters will be identified as such. Thus, the set B will only contain cheaters, and the dealer will pass the protocol. Moreover, the honest players' shares of $\mathcal{H}_0$ will also be preserved, so $\mathcal{H}_0$ will remain in $W_C^{(q)}$. □

2.1.2 Soundness

Lemma 2.2. *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 will employ a "quantum to classical" reduction, as in [LC99].

Lemma 2.3. *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 $\mathcal{M}_\ell$.

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 2.3 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 [CCD88], modified so as not to require the involvement of the dealer beyond the sharing stage).

Lemma 2.4 (Soundness of Modified Blobs from [CCD88]). *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.

Proof: Note that this statement is the same as Pr(the players in A do not have consistent shares *and* the dealer was not caught)$< 2^{n-k}$.

Recall that the adversary is adaptive, and can choose which set of players to corrupt on the fly. Nonetheless, the adversary's strategy can be reduced to choosing the set A of players who will be neither corrupted ($\in C$) nor accused ($\in B$), but such that $\mathbf{w}_0$ is not consistent on A, while the broadcast vectors $\mathbf{w}_\ell + b_\ell \mathbf{w}_0$ are all consistent.

Fix any particular set A. If the shares of $\mathbf{w}_0$ are not consistent on A, then there is at most a single value $b_\ell \in F$ such that the shares of $\mathbf{w}_\ell + b_\ell \mathbf{w}_0$ broadcast by players in A will be consistent, since the set of consistent vectors is a subspace. Thus, the probability of the dealer passing the tests with that set A is at most $\frac{1}{|F|^k}$. Overall, there are at most 2^n choices for the subset A, and so the adversary's total probability of being able to find a subset A of honest players for which cheating is possible is bounded above by $\frac{2^n}{|F|^k} \leq 2^{n-k}$. □

This completes the proof of Lemma 2.2.

Remark 4. As mentioned in the Definitions, we do not handle adaptive adversaries explicitly in this thesis. However, we believe that our protocols are secure against an adaptive adversary, and the previous proof gives some flavor of how the classical arguments can be used. In this case, a union bound argument was sufficient. For proving the security of the quantum protocols, a more sophisticated version of the quantum-to-classical reduction above (Lemma 2.3) would be necessary (and, we believe, sufficient).

2.1.3 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. $\sum_{\mathbf{w}\in W^\perp} |\mathbf{w}\rangle$).

"One-Level" Sharing and VQSS **for** $t < n/8$ Now suppose that there is some other code V such that before the protocol begins, all the systems $\mathcal{H}_0, \ldots, \mathcal{H}_k$ are in $V_{\tilde{B}}^{(q)}$. Then that property will not be affected by the protocol since the addition gates will not affect it. Thus, at the end of the protocol the shares of $\mathcal{H}_0$ would be in $\mathcal{C}_{\tilde{B}} = V_{\tilde{B}}^{(q)} \cap \mathcal{F}^{\otimes n} W_{\tilde{B}}^{(q)}$.

This leads to a first pass at a quantum sharing protocol: Have the dealer distribute $k+1$ groups of $k+1$ systems. In each group, use k of the systems to prove that the remaining system lies in $V_{\tilde{B}}^{(q)}$ using the subspace projection protocol. Next, take the $k+1$ resulting systems, and use k of them to prove that one of them is also in $\mathcal{F}^{\otimes n} W_{\tilde{B}}^{(q)}$ using the "dual" protocol.

Intuitively, this combination of the subspace projection protocol and the dual protocol achieves VQSS when $t < n/8$: since both the sets of apparent cheaters and of real cheaters have size at most t, the protocol allows the dealer to guarantee that the shared state is in $\mathcal{C}_{\tilde{B}}$ where $|\tilde{B}| < n/4$. Since the decoding operator is well-defined on such states (Proposition 1.4), the dealer is essentially committed to a unique value regardless of any changes the players make subsequently.

In the next section, we extend the ideas of this section, combining them with the classical VSS protocol of [CCD88] to obtain a VQSS protocol which is secure for $t < n/4$. We also show how to prove equivalence of that protocol to the ideal-model protocol of Section 1.2.6.

2.2 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 [CCD88] as well as on the literature on quantum fault-tolerance and error-correction, most notably on [AB99].

We first define the classical notion of "correctness" of a sharing used in [CCD88], and give a modified version of the [CCD88] VSS protocol that does not require the dealer's participation. We then describe our VQSS protocol (Section 2.2) and prove its security (Section 2.2.4–Section 2.2.6). Finally, we state the round and communication complexity of our protocol (Section 2.2.7) and some additional useful properties of the sharings it generates (Section 2.2.8).

2.2.1 Sharing Shares: 2-GOOD Trees

In the VSS protocol of [CCD88], 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 player j 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. Thus player j holds the j^{th} leaf in each branch of this tree.

We will run a cut-and-choose protocol similar to the subspace projection protocol above, 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.

N.B.: Since the dealer is one of the players in the protocol, we can in fact identify B with B_i, where the dealer is player i. However, by ignoring this fact we lose no correctness and gain some simplicity in the exposition and security proof of the protocol.

At the end of the protocol, we want to guarantee certain invariants:

Definition 4 (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 $i \notin C$ (corresponding to an honest player), we have $B_i \subseteq C$, i.e. all apparent cheaters are really 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 polynomial of degree $\leq d$, i.e. 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.

Why is this a useful property to guarantee? It turns out that this ensures the soundness of a sharing protocol. Suppose that all players broadcast their shares of a given 2-GOOD tree. Call the vector of shares in the i^{th} branch $\mathbf{v}_i$, so that player j holds the values $\mathbf{v}_i(j)$ for all i. Consider the reconstruction procedure Recover (Figure 2-2).

Algorithm 1. Recover$(T, V, B, B_1, ..., B_n)$
Input: a tree T which is 2-GOOD with respect to the code V and the sets $B, B_1, ..., B_n$.
Output: $a \in F$

1. For each branch $i \notin B$: Let $b = |B_i|$. If i is honest, then we expect the truncated word $\mathbf{v}_i|_{\bar{B}_i}$ to be within distance $t - b$ of a codeword in the truncated code $V|_{\bar{B}_i}$. Now this truncated code has distance $2t + 1 - b$: it can detect up to t errors and correct them when there are at most $t - b$ of them.

   If the truncated word $\mathbf{v}_i|_{\bar{B}_i}$ is at distance at most $t - b$ from a real codeword, then correct the error and let a_i be the interpolated value for that codeword. Otherwise output a null value $a_i = \perp$.
2. Take any set of $d + 1$ indices i such that $i \notin B$ and $a_i \neq \perp$. Find the unique polynomial p such that $p(i) = a_i$. Output $a = p(0)$ as the reconstructed secret.

Figure 2-2: Algorithm 1 (Reconstruction for a 2-GOOD tree)

Lemma 2.5. *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 (i.e. regardless of the values broadcast by the cheaters).*

We omit this proof here, since it is essentially re-proven in our analysis of the quantum protocol (see Lemma 2.11). We note that the protocols (and proofs) of [CCD88] used this lemma 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.

2.2.2 Classical VSS

Based on the discussion of the previous section, we give a modified version of the VSS protocol of [CCD88]. 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 Figure 2-3. Note that as before, the error-correcting code we use is V^{δ}, where $n = 4t + 1$ and $\delta = 2t$.

Remark 5. 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 (and is not made in [CCD88, RB89]). See Section 2.2.7 for further discussion.

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 2.6. *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 2.7. *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.*

2.2.3 VQSS Protocol

Given the previous protocol, and the observation that Subspace Projection can work simultaneously in both bases (Section 2.1.3), it is natural to attempt to run the classical VSS to check for errors in both bases. The resulting protocol is described in Figure 2-4 (Sharing Phase) and Figure 2-5 (Reconstruction Phase). 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. Note that we use the codes $V = V^{\delta} = V^{\delta'}$ and $W = W^{\delta} = W^{\delta'}$ (again 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 code is efficiently decodable.

Protocol 4 (Modified Classical VSS from [CCD88]). 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 $\mathsf{Recover}(T, V, B, B_1, ..., B_n)$.

Figure 2-3: Protocol 4 (Modified VSS protocol from [CCD88])

Protocol 5 (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_j}) 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) Updates sets B and $B_1, ..., B_n$ as in the classical VSS protocol.
  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-$X^{b'_j}$) 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) Updates sets B and $B_1, ..., B_n$ as in classical VSS protocol. Note that *for all* ℓ, we use code $W = V_0^{\perp}$.
       [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.

Figure 2-4: Protocol 5 (VQSS—Sharing Phase)

Protocol 6 (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'.

Figure 2-5: Protocol 6 (vqss—Reconstruction Phase)

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:

- Soundness: At the end of the protocol, if the dealer passes all tests then there is a unique state which will be recovered by the receiver, regardless of any changes made by the cheating players.
- Completeness (simplistic version): If the dealer is honest, then he will pass all tests and the state recovered by the receiver will be exactly the dealer's input system S.

At first, it may not be clear that the claim above for completeness is really sufficient, since it does not explicitly rule out the adversary learning any information about the secret system S. In fact, at some intuitive level it *is* sufficient, since any information the adversary was able to learn would cause a disruption of S (in general). Nonetheless, a formal proof of security requires a more sophisticated argument. We give the more formal proof, based on simulation, in Section 2.2.6.

2.2.4 (Informal) Soundness

Lemma 2.8. *The system has high fidelity to the following statement: "Either the dealer is caught 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: 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 2.2.2: 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-$\textsc{good}_V$ 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-$\textsc{good}_W$ in the computational basis. Since it is also 2-$\textsc{good}_V$ in the Fourier basis, the final rotation in Step 4 will leave it 2-$\textsc{good}_V$ in the computational basis and 2-$\textsc{good}_W$ in the Fourier basis. □

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.

Fact 2.9. *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: First, notice that these vectors are indeed pairwise orthogonal: for a pair of vectors, if any of the indices $j_i \in J$ differ for $i \geq 1$, we can distinguish the two states by measuring the syndrome of the i^{th} block of qubits. If none of the j_i differ but the indices j_0 differ, then we can distinguish the two states by correcting all the errors $E_{j_i}^{(i)}$, decoding the resulting blocks and measuring the syndrome of the final codeword. Finally, if the two states differ only by the choice of $a \in F$, we can distinguish them by correcting all errors, decoding and measuring the resulting qupit in the computational basis.

On the other hand, there are p^{n-1} choices for each of the $n+1$ indices $j_0, ..., j_n \in J$ and p choices for $a \in F$. Thus the total number of states is $\left(p^{n-1}\right)^{n+1} p = p^{n^2}$, and so the states must span all of $\mathbb{C}^{p^{n^2}}$. □

Proposition 2.10 (Characterizing 2-$\textsc{good}$ trees). *The space of trees of qupits which are 2-$\textsc{good}_V$ in the computational basis and 2-$\textsc{good}_W$ in the Fourier basis is spanned by the states*
$E_{j_1}^{(1)} \cdots E_{j_n}^{(n)} \mathcal{E}^{\otimes n} E_{j_0} \mathcal{E}|a\rangle$ *where*

- E_{j_0} *(or something in its equivalence class) acts only on B and*
- *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.)*

Proof: 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 2.9). 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. □

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.

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}. This will yield $n-2t$ qupits total. Applying the interpolation circuit again, we extract some system S which we take to be the output of the ideal interpolation. 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 $\mathcal{E}|a\rangle \longmapsto |a\rangle \otimes \mathcal{E}|0\rangle$.

Lemma 2.11. *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.*

Note that this is not necessarily true for a "one level" sharing (Section 2.1.3), unless $t < n/8$: by entangling errors with the shared data, the cheaters could arrange things so that more than t errors are detected only for certain possible values of the data, creating an entanglement between the data and the success or failure of the recovery.

Proof (of Lemma 2.11): 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.

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 2.10. 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 2.10.

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:

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

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 $|0\rangle$:

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

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:

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

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

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

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. □

Lemma 2.8 and Lemma 2.11 together imply that there is essentially a unique state which will be recovered in the reconstruction phase when the receiver R is honest. Thus, the Protocol 5 is sound, in the informal sense of Section 2.2.3.

2.2.5 (Informal) 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.

Lemma 2.12. *When the dealer D is honest, the effect of the verification phase on the shares which never pass through cheaters' hands is the identity.*

Proof: This follows essentially by inspection: for any codeword $\mathbf{v}$ of a linear code W, applying a controlled addition to $|\mathbf{v}\rangle \otimes \sum_{\mathbf{w}\in W} |\mathbf{w}\rangle$ results in the identity. Since this operation is transversal, the shares which never go through cheaters' hands will behave as if the identity gate was applied. □

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 go through honest players' hands only 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 2.13. *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 (Section 1.3.2) , 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.

2.2.6 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 2), 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 2 (p. 22).

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.

Simulation Outline

We give a sketch of the simulation procedure in Algorithm 2 (Figure 2-6).

Why does this simulation work?

- When D is cheating:
  - 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 2.11: 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 2.8), 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.

Algorithm 2. Simulation for VQSS (Protocol 5)

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 5, 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 5, 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.

Figure 2-6: Algorithm 2 (Simulation for VQSS)

We have essentially proved:

Theorem 2.14. *Protocol 5 is a statistically secure implementation of verifiable quantum secret sharing (Protocol 2).*

2.2.7 Round and Communication Complexity

In this section we show how to reduce the complexity of the protocol. For now, we will continue to assume that all public coins are generated using classical VSS: all players commit to a random value, then open all their values and take the sum to be the public coin. We discuss removing this assumption below.

Reducing the Number of Ancillas The first observation is that with these cut-and-choose protocols, it easy to check many trees at once for 2-GOOD-ness, so long as they were all generated by the same dealer. Suppose that we want to verify ℓ trees of quantum shares for 2-GOOD-ness in a certain basis. The dealer distributes the trees, and then creates k sharings of the ancilla state $\sum |a\rangle$ (as in the original protocol). In the original protocol, for each ancilla we chose a random coefficient $b \in F$ and performed the gate $(x, y) \mapsto (x, y + bx)$. In the new protocol, we add a random linear combination of all ℓ states to be checked into the ancilla: each challenge consists of ℓ coefficients $b_1, ..., b_\ell$ chosen publicly at random. We apply the linear gate $(x_1, ..., x_\ell, y) \mapsto (x_1, ..., x_\ell, y + \sum b_j x_j)$ to the ℓ trees and the ancilla. The resulting state is then measured in the computational basis and all players broadcast their shares.

To ensure good soundness, we can run this protocol k times in parallel, i.e. using k different ancillas and $k \cdot \ell$ random coefficients (i.e. k challenges of ℓ coefficients). Essentially the same analysis as in the previous sections shows that at the end of this protocol (with high fidelity) the dealer will have been caught or the shared states will *all* be 2-GOOD in the computational basis.

We can use this observation to improve the efficiency of our VQSS protocol. The dealer shares his secret S and also shares $2k$ ancillas. He uses the first k ancillas to check both the target state and the remaining k ancillas for consistency in the Fourier basis. He then uses the remaining ancillas to check the target state in the computational basis. The number of ancillas now scales linearly (instead of quadratically) but the protocol still requires a quadratic number of public values.

Generation of Public Values In the preceding discussion we assumed that public values were truly random. Such truly random coins can be implemented in our model using classical VSS, but in fact they need not be. As pointed out in [CCD88, RB89], it is sufficient to have players take turns generating challenges.

Suppose that each player broadcasts $\frac{k}{n}$ random challenges, and all players apply the challenge and measure and broadcast the result, as before. Then we are guaranteed that at least $k' = k\frac{n-t}{n}$ challenges will be chosen truly at random. Thus, by increasing k by a factor of $\frac{n}{n-t}$ we get the same soundness as before, and avoid expensive VSS protocols.

The final protocol takes three rounds, two of which use the broadcast channel. Each player sends and receives $kn\log|F|$ qubits. Moreover, the broadcast channel gets used k times to send challenges of (roughly) $k\log|F|$ bits. It is also used to broadcast k responses of $n\log|F|$ bits. To have soundness ϵ, we must have the number of truly random challenges be $k' \geq \frac{n+O(\log n)+log(1/\epsilon)}{\log|F|}$.

Since $\frac{n}{n-t}$ is constant, we get quantum communication complexity $O\left(\frac{(n+\log\frac{1}{\epsilon})^2}{n\log|F|}\right)$ per player and overall broadcast complexity $O\left((n+\log\frac{1}{\epsilon})(n+\frac{n+\log\frac{1}{\epsilon}}{n\log|F|})\right)$. This is optimized when each player broadcasts only a single challenge, i.e. $\log|F| = \frac{n+\log\frac{1}{\epsilon}}{n}$. In that case, we get quantum communication complexity $O(n+\log\frac{1}{\epsilon})$ per player and overall broadcast complexity $O\left(n(n+\log\frac{1}{\epsilon})\right)$.

2.2.8 Additional Properties of Two-Level Sharing

Level 2 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 $B, B_1, ..., B_n$ across various invocations of the protocol for the same dealer. Because set B_i corresponds to the set of players which player i has accused of cheating, we may take these sets as cumulative, and simply declare that a player is cheating whenever the union of all the set B_i (for the same i) is greater than t. Similarly for the set B. Thus, in the discussion below we assume that the sets $B, B_1, ..., B_n$ are the same for all invocations with a particular dealer.

1. Say the systems $S_{i,j}$, $S'_{i,j}$ form valid two-level sharings of states ρ, ρ' respectively (where S_{ij} corresponds to player j's share of branch i).

   Then applying the linear operation $(x,y) \to (x, y+bx)$ to the systems $S_{i,j} \otimes S'_{i,j}$ results in valid two-level sharings of the states obtained by applying the gate to the state $\rho \otimes \rho'$.

   In other words, if we denote the reconstruction procedure by $\mathcal{R}$ and the controlled-addition by c-X^b, we get that

   $$(c\text{-}X^b)\mathcal{R}^{\otimes 2} = \mathcal{R}^{\otimes 2}(c\text{-}X^b)^{\otimes n^2}$$

   (at least when restricted to the subspace of valid sharings).

2. Say the systems $S_{i,j}$ form valid two-level sharings of state ρ with respect to the codes V, W. Then applying $\mathcal{F}$ to each of the shares results in a valid sharing of the state $\mathcal{F}\rho\mathcal{F}^\dagger$ with respect to the codes W, V.

   That is, if $\mathcal{R}_{V,W}$ is the reconstruction procedure which uses code V in the computational basis and W in the Fourier basis, then when we restrict to the subspace of valid sharings we get:

   $$\mathcal{F}\mathcal{R}_{V,W} = \mathcal{R}_{W,V}\mathcal{F}^{\otimes n^2}$$

3. If all players measure their shares in a valid sharing of ρ and then apply classical reconstruction, then they will obtain the same result as if they had sent their shares to an honest reconstructor and asked him to broadcast the result of measuring ρ.
4. 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 $\sum_a |a\rangle$ by ensuring that all ancillas used for verification in the Fourier basis are in state $|0\rangle$, and again asking players to verify that the reconstructed codeword at the top level interpolates to 0 for the checks in the Fourier basis.

This last point is worth stressing: by tailoring the protocol, the dealer can verifiably share states $|0\rangle$ and $\sum_a |a\rangle$. This will be useful for sharing ancillas in the multi-party computation protocol.

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

Lemma 2.15. *No* VQSS *scheme exists for 4 players which tolerates one cheater.*

Before proving this, we need a result from quantum coding theory, on the relation between error-correction and erasure-correction:

Fact 2.16 (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.

It's an interesting and useful property of quantum information that it cannot be cloned, i.e. there is no procedure which takes an arbitrary, unknown pure state $|\psi\rangle$ and replaces it with two exact copies $|\psi\rangle \otimes |\psi\rangle$ (see Section 1.3.2). A corollary of this 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.

Proof (of Lemma 2.15): 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 that point, their joint state can be thought of as a (possibly mixed-state) encoding of the secret that was shared. In particular, an honest "receiver" Ruth, if she were given access to the state of all players, must be able to recover the shared state. Moreover, she must be able to do so even if one player suddenly decides to start cheating and introduces arbitrary errors into his 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. □

Corollary 2.17. *No* VQSS *scheme exists tolerating an adversary structure that contains four sets which cover all players.*

Proof: Suppose there exist four disjoint sets A, B, C, D such that $A \cup B \cup C \cup D = P$, and a VQSS scheme tolerating any adversary that can corrupt any one of those sets. Then we can construct a four player protocol tolerating one cheater by having each player simulate the players in one of the four sets. □

The optimality of our VQSS scheme is also an immediate corollary:

Theorem 2.18. *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 $t \geq n/4$.

2.4 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".

2.4.1 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 we have the simple, nearly-transversal fault-tolerant procedures of Section 1.3.3. 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 7 (Figure 2-7). 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

Protocol 7 (Top-Level Sharing). Dealer D takes as input a qupit S to share.

- **Sharing**

  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 4 in Section 2.2.8)

     (c) For $i = 1, ..., n - \delta - 1$:
     Runs level 2 sharing protocol to share state $|0\rangle$ (see Remark 4 in Section 2.2.8)

     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 now encodes the i-th component of an encoding of the input system 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.

Figure 2-7: Protocol 7 (Top-Level Sharing)

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.

Lemma 2.19. *At the end of Step 2, the system has high fidelity to "either the dealer is caught or measuring all n trees in the computational (resp. Fourier) basis yields a forest of n 2-*GOOD$_V$ *(resp. 2-*GOOD$_W$*) trees whose implicitly defined classical values $v_1, ..., v_n$ lie in V (resp. W).*

Proof: This follows from the linearity of the sharings generated by the VQSS scheme. □

Corollary 2.20 (Soundness of Top-Level Protocol). *At the end of the sharing phase (i.e. after Step 3), the system has high fidelity to "either the dealer is caught or the n shares of players $S_1, ..., S_n$ lie in $\mathcal{C}_C$".*

Proof: This is because the "rolling back" of the shares (i.e reconstruction of their respective components by all players) preserves measurement statistics in both bases. □

Lemma 2.21 (Completeness of Top-Level Protocol). *When D is honest, on input a pure state $|\psi\rangle$, the shared state will lie in* span $\{\mathcal{E}|\psi\rangle\}_C$*, i.e. will differ from an encoding of ψ only by a local operation on the cheaters' shares.*

Notice that the dealer can also prove to all players that he has shared a $|0\rangle$ state by simply proving that the system he is placing in the input position is in state $|0\rangle$.

Simulatability and Ideal Secret Sharing The top-level protocol (Protocol 7) is a simulatable VQSS protocol, just as was the original protocol. As before, the idea is that there is no perceivable difference between (a) running the protocol on input $|0\rangle$ and having the simulator "swap in" the real shared system S and (b) running the protocol honestly.

However, the top-level protocol is also a simulatable implementation of a different (and stronger) one-phase ideal task, which we call "ideal secret sharing" (Figure 2-8). In it, the dealer D sends his system S to the $\mathcal{TTP}$, and the $\mathcal{TTP}$ encodes it using the quantum error-correcting code $\mathcal{C}$ and sends the i-th component to player i.

The details of the simulation are substantially similar to those of Section 2.2.6. We get:

Theorem 2.22. *The top-level protocol (Protocol 7) is a statistically secure real-world implementation of ideal secret sharing (Protocol 8), for any $t < n/4$ (and thus in particular for $t < n/6$).*

2.4.2 Distributed Computation

Given the protocol of the previous section, and given the FTQC techniques described in Section 1.3.3, there is a natural protocol for multi-party computation of a circuit:

Protocol 8 (Ideal Secret Sharing). Input: Dealer D gets a qupit S.

1. D sends the $|F|$-dimensional system S to $\mathcal{TTP}$. If D fails to do this, $\mathcal{TTP}$ broadcasts "D is cheating" to all players.
2. $\mathcal{TTP}$ encodes D in $\mathcal{C}$. That is:
   (a) $\mathcal{TTP}$ creates δ states $\sum_a |a\rangle$ and $n - \delta - 1$ states $|0\rangle$.
   (b) $\mathcal{TTP}$ runs the linear encoding circuit (given by the $n \times n$ Vandermonde matrix) on S and the $n-1$ ancillas.
3. $\mathcal{TTP}$ sends the i^{th} component of the encoding to Player i.
4. For all i: Player i outputs either the qupit received from $\mathcal{TTP}$ or the message "D is cheating".

Figure 2-8: Protocol 8 (Ideal Secret Sharing)

have all players distribute their inputs via the top-level sharing (Protocol 7); apply the gates of U one-by-one, using the (essentially) transversal implementation of the gates described in Section 1.3.3; then have all players send their share of each output to the appropriate receiver. For completeness, we give this protocol in Figure 2-9 (p. 59).

One difficulty in the analysis of this protocol is the measurement results which are broadcast in the computation phase during Degree Reduction. 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. A full proof of security is somewhat tedious; instead, we sketch the main ideas in the remainder of this section.

Lemma 2.23. *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.*

Proof: Any state in $\mathcal{C}_C$ can be written as a mixture of linear combinations of basis states $E_j\mathcal{E}|\psi\rangle$ (see Lemma 1.5). The works on fault-tolerant computing show that the above procedures work correctly on such basis states. More importantly, they

Protocol 9 (Multi-party Quantum Computation).

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). The circuit they agree on should only use gates from the universal set in Section 1.3.3.

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

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 7), except this time with the one-dimensional code $\mathrm{span}\{\mathcal{E}_{\mathcal{C}}|0\rangle\}$ (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 7), with the one-dimensional code $\mathrm{span}\{\mathcal{E}_{\mathcal{C}^\delta}|0\rangle\}$ or $\mathrm{span}\{\mathcal{E}_{\mathcal{C}^{\delta'}}|0\rangle\}$, 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 1.3.3. 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 $|0\rangle$.

Figure 2-9: Protocol 9 (Multi-party Quantum Computation)

produce no new entanglement: the only opportunity to do so would come from the interaction in the measurement step of Degree Reduction. However, the resulting leftover ancilla is independent of the data in the computation, and hence provides no new information or entanglement. □

Theorem 2.24. *For any circuit U, Protocol 9 is a statistically secure real-world implementation of multi-party quantum computation (Protocol 1) as long as $t < n/6$.*

Proof: 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. □

Chapter 3

Open Questions

We conclude briefly with some open questions based on this research:

- Perhaps the most obvious question, given the results of this thesis, is to determine the true threshold for multi-party quantum computing, i.e. is it possible to tolerate up to $\lfloor (n-1)/4 \rfloor$ cheaters? We conjecture that it can indeed be done, but the techniques we use here are clearly not sufficient.

  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. The best known procedure for that code is a straightforward generalization of Shor's procedure for binary CSS codes [Sho96, AB99]. However, there is one point in that procedure at which *at most one* error can be tolerated. Such a procedure will fail when $t = \delta/2$ errors can be placed adversarially.

- A more subtle question is whether or not it is possible to remove the error probability from the protocols for verifiable quantum secret sharing. Given an error-free implementation of Ideal Secret Sharing, error-free multi-party computation is easy. However, attaining error-free VQSS seems difficult. Although we tried and failed to adapt the error-free classical techniques of Ben-Or, Goldwasser and Wigderson [BGW88], we conjecture that it is nonetheless possible to achieve error-free quantum computation.

- A potentially much more difficult question is what tasks are achievable when we allow cheating players to force the abortion of the protocol. That is, extend the ideal model so that the cheaters can, at any time, simply ask the trusted third party to stop the protocol entirely. In that setting VQSS becomes largely irrelevant since an essential aspect of VQSS is that the honest players be able to reconstruct the secret without the cheaters help. Thus, the bound of $n/4$ no longer seems hard; in fact, we conjecture some improvement is possible, possibly even up to tolerating any minority of cheating players.

Appendix A

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 A.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 A.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. $\Box$

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 A.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 $ST_B(\mathcal{C})$. $\Box$

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 $N_B(\mathcal{C})$.

Correspondence to an Idealized Experiment One interesting property of $ST_B(\mathcal{C})$ is that it is exactly the set of states which will arise in an idealized experiment in which cheaters introduce errors which are entangled with the data. Specifically, allow the cheaters to choose an arbitrary joint state $|\psi\rangle$ for two systems L and Aux (L is the logical data, Aux is auxiliary workspace). Now encode L using $\mathcal{C}$, and allow the cheaters to apply any operator which affects only Aux and the components of the encoding contained in B. Finally, trace out Aux so that only the components of the (corrupted) codeword are left.

Proposition A.4. *The set of possible states of the corrupted codeword system in the previous experiment is exactly $ST_B(\mathcal{C})$.*

Proof: We can assume w.l.o.g. that the adversary provides a pure state as input, since we can always purify the state with an ancilla and have him simply ignore the ancilla. Now, we are in the situation of considering $N_{B \cup Aux}^{pure}(\mathcal{C}')$, where $\mathcal{C}'$ is the code consisting of $\mathcal{C}$ when restricted to the codeword positions (and no restrictions on the Aux). By Proposition A.1 this is equal to $ST_{B \cup Aux}^{pure}(\mathcal{C}')$. But we have $ST_B(\mathcal{C}) = ST_{B \cup Aux}^{pure}(\mathcal{C}')$, i.e. once we trace out everything but A, there is no difference between $\mathcal{C}$ and $\mathcal{C}'$. $\Box$

Bibliography

[AB97] Dorit Aharonov and Michael Ben-Or. Fault tolerant quantum computation with constant error. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 176–188, El Paso, Texas, 4–6 May 1997. This is a preliminary version of [AB99].

[AB99] Dorit Aharonov and Michael Ben-Or. Fault tolerant quantum computation with constant error rate. Los Alamos eprint quant-ph/9906129. Journal version of [AB97] (submitted to SIAM J. Comp.), June 1999.

[ACM88] *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, Chicago, Illinois, 2–4 May 1988.

[Amb01] Andris Ambainis. A new protocol and lower bounds for quantum coin flipping. In *42nd Annual Symposium on Foundations of Computer Science*, Las Vegas, Nevada, October 2001. IEEE.

[ATVY00] Dorit Aharonov, Amnon Ta-Shma, Umesh V. Vazirani, and Andrew C. Yao. Quantum bit escrow. pages 705–714, 21–23May 2000.

[BB84] Charles Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–?, Bangalore, India, 1984. IEEE.

[BBP$^+$96] Charles Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumaker, John Smolin, and William Wooters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Physical Review Letters*, 76:722–725, 1996. Available as e-print quant-ph/9511027.

[BCG$^+$01] Howard Barnum, Claude Crépeau, Daniel Gottesman, Alain Tapp, and Adam Smith. Authentication of quantum messages. Manuscript, 2001.

[BCJL93] Gilles Brassard, Claude Crépeau, Richard Jozsa, and Denis Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *34th Annual Symposium on Foundations of Computer Science*, pages 362–371, Palo Alto, California, 3–5 November 1993. IEEE.

[BCMS98] Gilles Brassard, Claude Crépeau, Dominic Mayers, and Louis Salvail. Defeating classical bit commitments with a quantum computer. quant-ph/9806031, June 1998.

[Bea89] Donald 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 August 1989.

[Bea91] Donald Beaver. Foundations of secure interactive computing. In Feigenbaum [Fei91], pages 377–391.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In ACM [ACM88], pages 1–10.

[Can00] Ran Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[Can01] Ran 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.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In ACM [ACM88], pages 11–19.

[CDD$^+$99] Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations with dishonest minority. In Jacques Stern, editor, *Advances in Cryptology—EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*. IACR, Springer-Verlag, 1999.

[CDD$^+$01] Ran Canetti, Ivan Damgrd, Stefan Dziembowski, Yuval Ishai, and Tal Malkin. On adaptive vs. non-adaptive security of multiparty protocols. In *Advances in Cryptology—EUROCRYPT 2001* [IAC01], pages 262–279.

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *Proceedings of the Twenty-Eigth Annual ACM Symposium on the Theory of Computing*, pages 639–648, 1996.

[CGL99] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83:648–651, 1999.

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer Science*, pages 383–395, Portland, Oregon, 21–23 October 1985. IEEE.

[CGS01] Claude Crépeau, Daniel Gottesman, and Adam Smith. Verifiable quantum secret sharing and multi-party quantum computation. Manuscript, 2001.

[Cha00] H. F. Chau. Quantum-classical complexity-security tradeoff in secure multiparty computations. *Physical Review A*, 61, March 2000.

[CLS01] Claude Crépeau, Frédéric Légaré, and Louis Salvail. How to convert the flavor of a quantum bit commitment. In *Advances in Cryptology—EUROCRYPT 2001* [IAC01], pages 60–77.

[CS96] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1106, 1996.

[DEJ$^+$96] David Deutsch, Artur Ekert, Richard Jozsa, Ciara Macchiavello, Sandu Popescu, and Anna Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Physical Review Letters*, (77):2818–2821, 1996. Erratum: ibid. 80 (1998) 2022-2022. Also Los Alamos e-print quant-ph/9604039.

[DM00] Yevgeniy Dodis and Silvio Micali. Parallel reducibility for information-theoretically secure computation. In Mihir Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 74–92. IACR, Springer, 2000.

[DMS00] Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. volume 1807 of *Lecture Notes in Computer Science*, pages 300–315. IACR, Springer, 2000.

[Fei91] J. Feigenbaum, editor. *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*. IACR, Springer-Verlag, 1992, 11–15 August 1991.

[GC99] Daniel Gottesman and Isaac Chuang. Quantum teleportation is a universal computational primitive. *Nature*, November 1999.

[GL90] Shafi Goldwasser and Leonid 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 August 1990.

[GMW87] Oded Goldreich, Silvio Micali, and Avi 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.

[Got] Daniel Gottesman. Personal communication. 2001.

[Got00] Daniel Gottesman. On the theory of quantum secret sharing. *Physical Review A*, 61, 2000.

[IAC01] IACR. *Advances in Cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[LC96] Hoi-Kwong Lo and H. F. Chau. Why quantum bit commitment and ideal quantum coin tossing are impossible. In *Proceedings of the Fourth Workshop on Physics and Computation*, page 76, New England Complex Sys. Inst., 1996. Also available as quant-ph/9605026.

[LC97a] Hoi-Kwong Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410–3413, April 1997. Updated version available at quant-ph/9603004.

[LC97b] Hoi-Kwong Lo and H. F. Chau. Making an empty promise with a quantum computer. quant-ph/9709053. Also published in *Fortschritte der Phys.*, October 1997.

[LC99] Hoi-Kwong Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283(5410):2050–2056, 26 March 1999.

[Lyn96] Nancy Lynch. *Distributed Algorithms*, chapter 6. Addison-Wesley, 1996.

[May96] Dominic Mayers. The trouble with quantum bit commitment. quant-ph/9603015, March 1996.

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78:3414–3417, 1997. quant-ph/9605044.

[May98] Dominic Mayers. Unconditional security in quantum cryptography. quant-ph/9802025, Updated September 1998.

[MR91] Silvio Micali and Phillip Rogaway. Secure computation (abstract). In Feigenbaum [Fei91], pages 392–404.

[MS99] Dominic Mayers and Louis Salvail. Unconditionally secure quantum coin flipping. quant-ph/9806031, April 1999.

[NC00] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[PW00] Birgit Pfitzmann and Michael Waidner. Composition and integrity preservation of secure reactive systems. In *ACM Conference on Computer and Communications Security 2000*, pages 245–254, 2000.

[RB89] Tal Rabin and Michael 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.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[Sho96] Peter W. Shor. Fault-tolerant quantum computation. In *37th Annual Symposium on Foundations of Computer Science*, pages 56–65, Burlington, Vermont, 14–16 October 1996. IEEE.

[SP00] Peter W. Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85:441–444, 2000.

[Ste96] Andrew Steane. Simple quantum error correcting codes. *Physical Review A*, 54, 1996.

[vdG97] Jeroen van de Graaf. *Towards a formal defintion of security for quantum protocols*. PhD thesis, Université de Montréal, Canada, December 1997. Available from http://www.cenapad.ufmg.br/~jvdg/.