Secure Multi-party Quantum Computation

Claude Crépeau*
McGill University
crepeau@cs.mcgill.ca

Daniel Gottesman†
UC Berkeley
gottesma@eecs.berkeley.edu

Adam Smith‡
MIT
asmith@theory.lcs.mit.edu

ABSTRACT

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

Keywords

Quantum cryptography, multi-party protocols, secure function evaluation, distributed computing

1. INTRODUCTION

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

We investigate a quantum version of an extensively studied classical problem, *secure multi-party computation* (or *secure function evaluation*), first introduced by [14]. 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 of the input states. The output of the circuit is broken into n components $\mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_n$, and P_i receives the output $\mathcal{H}_i$. Note that the inputs to this protocol are arbitrary quantum states—the player providing an input need only have it in his possession; he does not need to know a classical description of it. Moreover, unlike in the classical case, we cannot assume without loss of generality that the result of the computation will be broadcast. Instead, each player in the protocol receives some part of the output.

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

Verifiable Quantum Secret Sharing. In order to construct MPQC protocols, we consider a subtask which we call *verifiable quantum secret sharing*. In classical cryptography, a verifiable secret sharing scheme [9] is a two phase protocol with one player designated as the "dealer". After the first phase (*commitment*), the dealer shares a secret amongst the players. In the second phase (*recovery*), the players reconstruct the value publicly.

The natural quantum version of this allows a dealer to share a state ρ (possibly unknown to him but nonetheless in his possession). Because quantum information cannot be cloned, 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 $\leq t$ players:
- *Soundness:* As long as R is honest and the dealer passes the commitment phase successfully, then there is a unique quantum state which can be recovered by R.
- *Completeness:* When D is honest, then he always passes the commitment phase. Moreover, when R is also honest, then the value recovered by R is exactly D's input ρ.
- *Privacy:* When D is honest, no other player learns info about D's input until the recovery step.

Note that for quantum data, the privacy condition is redundant: any information obtained about the shared

*Supported by Québec's FCAR and Canada's NSERC.
†Supported by the Clay Mathematics Institute.
‡Supported by the US DoD MURI program administered by the Army Research Office under grant DAAD19-00-1-0177.

state would imply some disturbance of that state, contradicting the completeness requirement.

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

Beyond these specific results, there are a number of conceptual contributions of this paper to the theory of quantum cryptographic protocols. We provide a simple, general framework for defining and proving the security of distributed quantum protocols in terms of equivalence to an ideal protocol involving a third party. This follows the definitions for classical multi-party protocols. 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 [8]. The new property our protocols have is that dealers do not need to remember the randomness they use when constructing shares to distribute to other players. This allows them to replace a random choice of coins with the *superposition* over all such choices.

1.1 Previous Work

Classical MPC. Multi-party computing was introduced by Goldreich, Micali and Wigderson [14], who showed that *under computational assumptions*, secure multi-party evaluation of any function was possible tolerating any minority of cheating players, i.e. if and only if $t < \frac{n}{2}$. If one assumes pairwise secure channels but no computational assumptions, then one can compute any function securely if and only if $t < n/3$ [5, 8]. If one further assumes the availability of a secure broadcast channel, then one can in fact tolerate $t < n/2$, and no more ([21, 3, 11]). All of these protocols rely on verifiable secret sharing as a basic tool. Our solution draws most heavily on the techniques of Chaum, Crépeau and Damgård [8].

Beyond these basic protocols, much work has focused on finding proper definitions of security, e.g. [15, 4, 18, 20, 6]. We adopt a simple definition based on the initial definitions of Canetti.

Quantum Secret Sharing. Relatively little work exists on multi-party cryptographic protocols with quantum data. Secret sharing with a quantum secret was first studied by Cleve et al. [10], who showed an equivalence with quantum error-correcting codes (QECC). Their scheme is the basis of our protocols. Chau [7] deals with classical computations, but also mentions the problem of verifiable quantum secret sharing as an open question.

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

One can view multi-party computation as fault-tolerant computing with a different error model, one that is suited to distributed computing. The MPQC model is weaker in some respects since we assume that errors will always occur in the same, limited number of positions, i.e. errors will only occur in the systems of the t corrupted players. In other respects, the error model of MPQC is stronger: in our setting errors may be *maliciously* coordinated. In particular, they will not be independently placed, and they may in fact depend on the data of the computation—the adversaries will use any partial information known about the other players' data, as well as information about their own data, to attempt to corrupt the computation. For example, several FTQC algorithms rely on the fact that at certain points in the computation, at most one error is likely to occur. Such algorithms will fail when errors are placed adversarially. Techniques from FTQC are nonetheless useful for multi-party computing. We will draw most heavily on techniques due to Aharonov and Ben-Or [2].

1.2 Definitions and Model

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

We assume that every pair of participants is connected by perfect (i.e. authenticated, unjammable, secret) quantum and classical channels, and that there is a classical authenticated broadcast channel to which all players have access. Because we will always consider settings where $t < \frac{n}{2}$, we can also assume that players can perform *classical* multi-party computations securely [11][1]. The adversary is an arbitrary quantum algorithm (or family of circuits) $\mathcal{A}$ (not necessarily polynomial time), and so the security of our protocols does not rely on computational assumptions.

The real and ideal models, as well as the notion of security, are specified more carefully in [22]. In this paper, we use the following informal specifications of

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

the ideal protocols. The real protocols are secure if they succeed in simulating the ideal ones.

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

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

1.3 Preliminaries

We present the notation necessary for reading the protocols and proofs in this paper. For a more detailed explanation of the relevant background, see [22] or a textbook such as [19].

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

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

Specifically, we will use quantum Reed-Solomon codes from [2]. We specify a quantum RS code by a single parameter $\delta < n/2$. The classical Reed-Solomon code V^δ is the set of all vectors $\hat{\mathbf{q}} = (q(1), q(2), \ldots, q(n))$, where q is any univariate polynomial 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: $V^\delta = \{\hat{\mathbf{q}} : q \in F[x] : \deg(q) \le \delta\}$ and $V_0^\delta = \{\hat{\mathbf{q}} : \deg(q) \le \delta \text{ and } q(0) = 0\} \subseteq V^\delta$. The code V^δ has minimum distance $d = n - \delta$, and an efficient error correction procedure. Let $\delta' = n - \delta - 1$. There are constants $d_1, ..., d_n \in \mathbb{Z}_p$ such that the dual of the code V^δ is just the code $V_0^{\delta'}$, rescaled by d_i in the i^{th} coordinate; similarly, the dual of V_0^δ is a rescaled version of $V^{\delta'}$. Denote these duals by $W_0^{\delta'}, W^{\delta'}$, respectively.

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

Transversal Operations. A nice result from fault-tolerant computing [2, 13] is that one can in fact perform many operations on data encoded by a quantum RS code using only local operations and classical information transmitted between the components. Consider the following gates:

1. Shift: $X^c : |a\rangle \mapsto |a+c\rangle$,
2. SUM: $(c\text{-}X) : |a, b\rangle \mapsto |a, a+b\rangle$,
3. Scalar multiplication: $0 \ne c \in F$, $S_c : |a\rangle \mapsto |ac\rangle$,
4. Phase Shift: $Z^c : |a\rangle \mapsto w^{ca}|a\rangle$,
5. Fourier Transform: $\mathcal{F}_r : |a\rangle \mapsto \frac{1}{\sqrt{p}} \sum_{b \in F} w^{rab}|b\rangle$,
6. Toffoli (Multiplication): $|a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|c + ab\rangle$.

These gates are universal [2], in the sense that a sequence of these gates can approximate any unitary operation with arbitrary accuracy. 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. For any CSS code, the gates 1 through 4 from the set above can be implemented *transversally*, that is using only local operations which affect the same component of two codewords. Measurement and the remaining two operations can be performed almost transversally.

Measurement. For a quantum RS code, measuring each component of the encoding of $|s\rangle$ yields a vector $\hat{\mathbf{q}} = (q(1), ..., q(n))$ where $q(0) = s$. 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, it can tolerate arbitrary corrupton of $\delta/2$ of the positions in the codeword if classical error correction is first applied to the vector of measurement results.

Fourier and Toffoli gates. For CSS codes, applying the Fourier transform transversally maps data encoded with the codes V, W to the Fourier transform of that data, encoded with the dual code $\tilde{\mathcal{C}}$ defined via the codes W, V. For quantum RS codes, rescaling each component of the dual code of $\mathcal{C}^\delta$ produces the code $\mathcal{C}^{\delta'}$. This allows one to perform the map $\mathcal{E}_{\mathcal{C}^\delta}|\psi\rangle \mapsto \mathcal{E}_{\mathcal{C}^{\delta'}}(\mathcal{F}|\psi\rangle)$, where $\mathcal{E}_\mathcal{C}$ is the encoding map for a code $\mathcal{C}$.

When $n = 2\delta + 1$, we have $\delta' = \delta$, so the Fourier transform is in fact transversal, but the Toffoli gate is difficult to perform.

When $n = 3\delta + 1$, neither the Fourier transform nor the Toffoli gate is transversal, but they can both be reduced to *degree reduction* via transversal operations [2]. Degree reduction maps an arbitrary state $|\psi\rangle$ encoded using $C^{\delta'}$ to $|\psi\rangle$ encoded with C^δ.

The circuit we use for degree reduction is due to Gottesman and Bennett [13]. We start with one block encoded using $\mathcal{C}^{\delta'}$ (system $\mathcal{H}_1$), and an ancilla block in the state $\mathcal{E}_{\mathcal{C}^\delta}(\sum |a\rangle)$ (system $\mathcal{H}_2$). Perform a SUM gate from $\mathcal{H}_2$ to $\mathcal{H}_1$ (this can be done transversally by a property of the codes $\mathcal{C}^\delta$). Measure $\mathcal{H}_1$ in the computational basis, obtaining b, and apply $X^b S_{-1}$ to $\mathcal{H}_2$. The system $\mathcal{H}_2$ now contains the data, encoded using $\mathcal{C}^\delta$. This entire procedure can be performed transversally except for the measurement step. However, as noted above, measurement requires only classical communication between the components.

2. NEIGHBORHOODS OF QUANTUM CODES

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

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

1. ρ differs from a state in $\mathcal{C}$ only by some quantum superoperator $\mathcal{O}$ acting only on B:
   $N_B(\mathcal{C}) = \{\rho : \exists\rho' \text{ in } \mathcal{C}, \exists\mathcal{O} \text{s.t. } \rho = \mathcal{O}(\rho')\}$.
2. ρ cannot be distinguished from a state in $\mathcal{C}$ by looking only at positions in A.
   $ST_B(\mathcal{C}) = \{\rho : \exists\rho' \text{ in } \mathcal{C} \text{ s.t. } \mathrm{Tr}_B(\rho) = \mathrm{Tr}_B(\rho')\}$.
3. Specifically for CSS codes, one can require that the state ρ pass checks on A in both bases, i.e. that measuring either the V_B syndrome in the computational basis, or the W_B syndrome in the Fourier basis, yields 0. The set of states which pass this test is: $\mathcal{C}_B = V_B^{(q)} \cap \mathcal{F}^{\otimes n} W_B^{(q)}$.

In general, these notions form a strict hierarchy: $N_B(\mathcal{C}) \subsetneq ST_B(\mathcal{C}) \subsetneq \mathcal{C}_B$. Only notion (3) is always a subspace (see [22] for details).

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

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

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

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

Proof: 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. However, 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 $E_j\mathcal{C}$. 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 $\mathrm{Tr}(P_j\rho) = 0$ 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. □

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

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

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

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

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

3. A TWO LEVEL VQSS PROTOCOL

In this section we define a two-tiered protocol for VQSS. It is based on the VQSS protocols of [8] as well as on the literature on quantum fault-tolerance and error correction, most notably on [2]. Detailed proofs for the claims of this section are in [22]. However, some intuition is given by the proofs of Section 2.

3.1 Sharing Shares: 2-GOOD Trees

In the VSS protocol of [8], the dealer D takes his secret, splits it into n shares and gives the i^{th} component to player i. Player i then shares this secret by splitting it into n shares and giving the j^{th} share to player j. Thus, there are n^2 total shares, which can be thought of as the leaves of a tree with depth 2 and fan-out n: each leaf is a share; the i^{th} branch corresponds to the shares created by player i, and the root corresponds to the initial shares created by the dealer. Player j holds the j^{th} leaf in each branch of this tree. We will run a cut-and-choose protocol in order to guarantee some kind of consistency of the distributed shares.

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

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

1. *For each $i \notin C$ (i.e., corresponding to an honest player), we have $B_i \subseteq C$, i.e. all apparent cheaters are real cheaters.*
2. *For each branch $i \notin B$, the shares held by the honest players* not *in B_i should all be consistent with some codeword in V, i.e. the vector of all shares should be in $V_{B_i \cup C}$, where C is the set of cheating players.*
   N.B.: Because there are at most t players in B_i and at most t cheaters, there are at least $d+1 \leq n-2t$ honest players remaining, and so the polynomial above is uniquely defined. This guarantees that for each branch $i \notin B$, there is a unique value $a_i \in F$ which is obtained by interpolating the shares of the honest players not in B_i.
3. *For $i \notin B$, the values a_i defined by the previous property are all consistent with a codeword of V (i.e. the vector $(a_1, ..., a_n)$ is in V_B).*

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

3.2 VQSS Protocol

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

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

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

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

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

Proof of this is via a "quantum-to-classical" reduction, similar to that of [17]. First, checks in the computational and Fourier bases don't interfere with each other, since they commute for CSS codes. Second, in a given basis, we can assume w.l.o.g. that all ancillae are first measured in that basis, reducing to a classical analysis similar to [8].

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

The *real* recovery operator $\mathcal{D}$ is given by Protocol 2. The following lemma then applies, following essentially from Proposition 2.2.

Lemma 3.2. *Given a tree of qupits which is 2-GOOD in both bases, the outputs of $\mathcal{R}^I$ and $\mathcal{D}$ are the same. In particular, this means that no changes made by cheaters to their shares can affect the outcome of $\mathcal{D}$.*

Lemmas 3.1 and 3.2 show that there is essentially a unique state which will be recovered in the reconstruction phase when the receiver R is honest.

Protocol 1 (vqss—Sharing Phase). *Dealer D gets as input a quantum system S to share.*

- ***Sharing:***
  1. *The dealer D prepares $(k+1)^2$ systems of n qupits each, called $S_{\ell,m}$ (for $\ell = 0, ..., k$ and $m = 0, ..., k$):*
     (a) *Encodes S using $\mathcal{C}$ in $S_{0,0}$.*
     (b) *Prepares k systems $S_{0,1}, ..., S_{0,k}$ in the state $\sum_{a\in F} \mathcal{E}_{\mathcal{C}}|a\rangle = \sum_{v\in V} |v\rangle$.*
     (c) *Prepares $k(k+1)$ systems $S_{\ell,m}$, for $\ell = 1, ..., k$ and $m = 0, ..., k$, each in the state $|\bar{0}\rangle = \sum_{v\in V_0} |v\rangle$.*
     (d) *For each of the $(k+1)^2$ systems $S_{\ell,m}$, D sends the i^{th} component (denoted $S^{(i)}_{\ell,m}$) to player i.*
  2. *Each player i, for each $\ell, m = 0, ...k$:*
     (a) *Encodes the received system $S^{(i)}_{\ell,m}$ using $\mathcal{C}$ into an n qupit system $S_{\ell,m,i}$.*
     (b) *Sends the j^{th} component $S^{(j)}_{\ell,m,i}$ to player j.*
- ***Verification:***
  1. *Get public random values $b_1, ..., b_k \in_R F$. For each $\ell = 0, ..., k$, $m = 1, ..., k$, each player j:*
     (a) *Applies the SUM gate (c-X^{b_m}) to his shares of the systems $S_{\ell,0,i}$ and $S_{\ell,m,i}$.*
     (b) *Measures his share of $S_{\ell,m,i}$ and broadcasts the result (i.e. each player broadcasts $k(k+1)n$ values).*
     (c) *For each $i \in \{1, ..., n\}$, players update the set B_i based on the broadcast values: there are $(k+1)kn$ broadcast words $\mathbf{w}_{\ell,m,i}$. Applying classical decoding to each of these yields min-weight error vectors $\mathbf{e}_{\ell,m,i}$ with support set $B_{\ell,m,i}$. Set $B_i = \cup_{\ell,m} B_{\ell,m,i}$. If there are too many errors, instead add i to the global set B.*
     (d) *Furthermore, players do the same at the root level: for all $i \notin B$, there is an interpolated value a_i which corresponds to the decoded codeword from the previous step. Players also decode the codeword $(a_1, ..., a_n)$ and update B accordingly (i.e. by adding any positions where errors occur to B).*
  2. *All players apply the Fourier transform $\mathcal{F}$ to their shares.*
  3. *Get public random values $b'_1, ..., b'_k \in_R F$. For $\ell = 1, ..., k$, each player j:*
     (a) *Applies the SUM gate (c-$X^{b'_\ell}$) to his shares of the systems $S_{0,0,i}$ and $S_{\ell,0,i}$.*
     (b) *Measures his share of $S_{\ell,0,i}$ and broadcasts the result (i.e. each player broadcasts kn values).*
     (c) *For each $i \in \{1, ..., n\}$, players update B_i and B based on the broadcast values (as in Step 1c). [Note: the sets B and $B_1, ..., B_n$ are cumulative throughout the protocol.]*
  4. *All players apply the inverse transform $\mathcal{F}^{-1}$ to their shares of $S_{0,0}$.*
- *The remaining shares (i.e. the components of the n systems $S_{0,0,i}$) form the sharing of the state ρ.*

Protocol 2 (vqss—Reconstruction Phase). *Player j sends his share of each of the systems $S_{0,0,i}$ to the receiver R, who runs the following decoding algorithm:*
1. *For each branch i: Determine if there is a set $\tilde{B}_i$ such that $B_i \subseteq \tilde{B}_i$, $|\tilde{B}_i| \leq t$ 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'.*

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

Consider the case where the dealer's input is a pure state $|\psi\rangle$. On one hand, we can see by inspection that an honest dealer will always pass the protocol. Moreover, since the shares that only go through honest players' hands remain unchanged, it must be that if some state is reconstructed, then that state is $|\psi\rangle$, 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 $|\psi\rangle$, an honest reconstructor will reconstruct $|\psi\rangle$. We have proved:

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

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

Simulatability. The claims above show that the protocol satisfies an intuitive notion of security. In this section we sketch a proof that the protocol satisfies a more formal notion of security: it is equivalent to a simple ideal model protocol. The equivalence is *statistical*, that is the outputs of the real and ideal protocols may not be identical but will have very high fidelity to one another.

THEOREM 3.4. *Protocols 1 and 2 are a statistically secure* VQSS *scheme.*

The ideal protocol is sketched in Section 1.2. To show equivalence, 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 experiencing the real protocol.

The idea is that the simulator will simulate the regular VQSS protocol either on input provided by a cheating dealer or on bogus data $|0\rangle$, and then extract and/or change the shared state as needed.

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

1. When D is cheating:
   (a) 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 $\mathcal{TTP}$. In the reconstruction phase, $\mathcal{S}$ gets the same state back from $\mathcal{TTP}$, and runs the interpolation circuit in reverse. Thus, the two executions of the interpolation circuit cancel out.
   (b) When R is honest, the faithfulness of the simulation comes from Lemma 3.2: in the real protocol, R outputs the result of the regular decoding operator. In the simulation, R gets the output of the ideal interpolation. Since the shared state has high fidelity to a 2-GOOD tree (by Lemma 3.1), the outputs will be essentially identical in both settings (i.e. they will have high fidelity).
2. When D is honest:
   (a) When R is also honest, the faithfulness of the simulation follows from the completeness and privacy properties of the real protocol. Privacy implies that the adversary $\mathcal{A}_1$ cannot tell that it is actually participating in a sharing of $|0\rangle$ rather than the dealer's state, and completeness means that R in the real protocol gets a state with high fidelity to that received by R in the ideal protocol.
   (b) When R is a cheater, $\mathcal{S}$ does not get S from $\mathcal{TTP}$ until the reconstruction phase. Then he applies the ideal interpolation circuit to extract the $|0\rangle$ state used during the verification phase, swaps S with $|0\rangle$, then runs the ideal interpolation circuit backwards. Since the ideal interpolation circuit only acts on shares of the honest players, $\mathcal{S}$ is capable of performing these operations without tipping off $\mathcal{A}_1$ to the fact that it is in a simulation. By the completeness property of the real protocol and the no-cloning theorem, the residual state left over after the ideal interpolation circuit (i.e., the state of the cheaters) has almost no correlation to the data shared using the circuit, so swapping in S and running the circuit backwards gives us a state with high fidelity to the state that would have resulted from sharing S directly with the same $\mathcal{A}_1$. Thus, the simulation is faithful in this case as well.

We have essentially proved Theorem 3.4.

3.3 Additional Properties

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

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

4. LOWER BOUND FOR VQSS

LEMMA 4.1. *No 4-player* VQSS *scheme tolerates one cheater.*

Proof: Suppose such a scheme exists. Consider a run of the protocol in which all players behave perfectly honestly until the end of the sharing phase, at which point one (unknown) player introduces an arbitrary error. However, an honest "receiver" Ruth, given access to the state of all players, must still be able to recover the shared state. Thus, the joint state of all players constitutes a four-component QECC correcting one error. However, no such code exists, not even a mixed-state one, by the quantum Singleton bound [16]. □

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

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

Note that we have only proved the impossibility of *perfect* VQSS protocols. However, the quantum Singleton bound still holds when exact equality is replaced by

Algorithm 1. *Simulation for* vqss *(Protocol 1)*

Sharing/Verification phase

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

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

Reconstruction phase

- *If R is a cheater, $\mathcal{S}$ receives the system S from $\mathcal{TTP}$. $\mathcal{S}$ runs the interpolation circuit backwards on the positions in I, with S in the position of the secret. $\mathcal{S}$ 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.

approximate correctness, and so in fact even statistical vqss schemes are impossible when $t \geq n/4$.

5. MULTI-PARTY COMPUTATION

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

5.1 Top-Level Sharing Protocol

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

The new scheme is given in Protocol 3. The idea is that the previous vqss scheme allows distributed computation of linear gates and Fourier transforms on states shared by the same dealer. It also allows verifying that a given shared state is either $|0\rangle$ or $\sum |a\rangle$. The players will use this to perform a distributed computation of the encoding gate for the code $\mathcal{C}$. Thus, the dealer will share the secret system S, as well as δ states $\sum |a\rangle$ and $n-\delta-1$ states $|0\rangle$. Players then apply the (linear) encoding gate, and each player gets sent all shares of his component of the output. As before, the main lemmas are soundness and completeness of the protocol:

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

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

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

5.2 Distributed Computation

Given the protocol of the previous section, and known fault-tolerant techniques, there is a natural protocol for secure multi-party computation of a circuit: have all players distribute their inputs via the top-level sharing (Protocol 3); apply the gates of U one-by-one, using the (essentially) transversal implementation of the gates described in Section 1.3; then have all players send their share of each output to the appropriate receiver. See Protocol 4.

Protocol 3 (Top-Level Sharing). *Dealer D takes as input a qupit S to share.*

- *1.* ***(Distribution)*** *The dealer D:*
  - *(a) Runs the level 2* VQSS *protocol on input S.*
  - *(b) For $i = 1, ..., \delta$: Runs level 2 sharing protocol to share state $\sum_a |a\rangle$ (see Remark 2 in Section 3.3)*
  - *(c) For $i = 1, ..., n - \delta - 1$: Runs level 2 sharing protocol to share state $|0\rangle$ (see Remark 2 in Section 3.3) Denote the n shared systems by $S_1, ..., S_n$ (i.e. S_1 corresponds to S, $S_2, ..., S_{\delta+1}$ correspond to $\sum_a |a\rangle$ and $S_{\delta+2}, ..., S_n$ correspond to $|0\rangle$). Note that each S_i is a two-level tree, and thus corresponds to n components in the hands of each player.*

  2. ***(Computation)*** *Collectively, the players apply the Vandermonde matrix to their shares of $S_1, ..., S_n$. (If D is honest then system S_i encodes the i-th component of an encoding of the input S).*

  3. For each i, all players send their shares of S_i to player i, who decodes them (as per Protocol 2).

- ***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 ($\mathcal{D}$ is the decoding algorithm for $\mathcal{C}$).

Protocol 4 (Multi-party Quantum Computation).

1. ***Input Phase:***

(a) For each i, player i runs Top-Level Sharing with input S_i.

(b) If i is caught cheating, then some player who has not been caught cheating yet runs Top-Level Sharing (Protocol 3), 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 3), 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. 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 $|0\rangle$.

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

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

Theorem 5.4. *For any circuit U, Protocol 4 is a statistically secure implementation of multi-party quantum computation 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 from the security of the top-level sharing protocol and the previous discussion on fault-tolerant procedures. $\square$

6. OPEN QUESTIONS

Given our results, the most obvious open question is if MPQC is possible when $n/6 \leq t < n/4$. Another natural direction of research is to find a VQSS protocol with zero error. For example, the techniques of [5] for the classical case do not seem to apply to the quantum setting. Finally, one can ask what tasks are achievable when we allow cheating players to force the abortion of the protocol (usually called an "optimistic protocol").

Acknowledgements

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

7. REFERENCES

[1] *Proc. of 20th STOC*, Chicago, Illinois, 2–4 May 1988.

[2] D. Aharonov and M. Ben-Or. Fault tolerant quantum computation with constant error rate. quant-ph/9906129. Preliminary version in *STOC '97*. Submitted to SIAM J. Comp., June 1999.

[3] D. Beaver. Multiparty protocols tolerating half faulty processors. In G. Brassard, editor, *Proc. of CRYPTO '89*, volume 435 of *LNCS*, pages 560–572. IACR, Springer-Verlag, 1990, 20–24 Aug. 1989.

[4] D. Beaver. Foundations of secure interactive computing. In Feigenbaum [12], pages 377–391.

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

[6] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. of FOCS 2001*, pages 136–147, 2001.

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

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

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

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

[11] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations with dishonest minority. In J. Stern, editor, *Proc. of EUROCRYPT 99*, volume 1592 of *LNCS*. IACR, Springer-Verlag, 1999.

[12] J. Feigenbaum, editor. *Proc. of CRYPTO '91*, volume 576 of *LNCS*. IACR, Springer-Verlag, 1992, 11–15 Aug. 1991.

[13] D. Gottesman and C. H. Bennett. Unpublished work. 1998.

[14] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proc. of 19th STOC*, pages 218–229, New York City, 25–27 May 1987.

[15] S. Goldwasser and L. A. Levin. Fair computation of general functions in presence of immoral majority. In A. J. Menezes and S. A. Vanstone, editors, *Proc. of CRYPTO '90*, volume 537 of *LNCS*, pages 77–93. IACR, Springer-Verlag, 1991.

[16] E. Knill and R. Laflamme. A theory of quantum error-correcting codes. *Physical Review A*, 55:900–911, 1997. quant-ph/9604034.

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

[18] S. Micali and P. Rogaway. Secure computation (abstract). In Feigenbaum [12], pages 392–404.

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

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

[21] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proc. of 21st STOC*, pages 73–85, Seattle, Washington, 15–17 May 1989.

[22] A. Smith. Multi-party quantum computation. Master's thesis, MIT, Aug. 2001. Available as quant-ph/0111030.