Secure Multi-party Quantum Computing

Claude Crépeau, McGill
Daniel Gottesman, UC Berkeley
Adam Smith, MIT
Preliminary version presented at NEC workshop on quantum crypto after QIP 2000

Since then:

• Protocols have changed a little.

• Definitions have been found.

• Proofs have changed a lot
Classical Distributed Protocols

• Extensively studied

• Many applications
  – Banking / E-commerce
  – Electronic Voting
  – Auctions / Bidding
Questions for Quantum Protocols

• Do existing protocols remain secure?
  – Not always: factoring, discrete log
Questions for Quantum Protocols

• Do existing protocols remain secure?

• Can we find better / more secure protocols for existing tasks?
  
  – E.g. Key distribution, coin flipping (?), “quantum voting”
Questions for Quantum Protocols

• Do existing protocols remain secure?

• Can we find better / more secure protocols for existing tasks?

• What new, quantum tasks can we perform?
  – E.g. Quantum Secret-Sharing, Zero-Knowledge, Authentication, Entanglement Purification
  – General trend: do cryptography with quantum data
  – Goal: building blocks for complex protocols
Overview

• What is multi-party (quantum) computing?
• A Sketch of the Protocol
• An Impossibility Result
What is Multi-party Computing?
Classical Multi-party Computing

- Network of \( n \) players
- Each has input \( x_i \)
- Want to compute \( f(x_1, \ldots, x_n) \) for some known function \( f \)
- \textit{E.g.} electronic voting
Classical Multi-party Computing

Even if $t$ out of $n$ players try to cheat:

1. Cheaters learn nothing (except output)
2. Cheaters cannot affect output
Classical Multi-party Computing

Even if \( t \) out of \( n \) players try to cheat:

1. Cheaters learn nothing (except output)
2. Cheaters cannot affect output

Even with unbounded computation time
Quantum Multi-party Computing

- Players’ inputs are quantum states
  - Possibly entangled
  - No description necessary
    (protocol is “oblivious”)
- Output is quantum
- Want to evaluate a known quantum circuit $U$
- Player $i$ gets $i$-th component of output
Quantum Multi-party Computing

- Players’ inputs form an arbitrary state $\rho$ in $H_1 \otimes H_2 \otimes \ldots \otimes H_n$
- Player $i$ holds $i$-th component:
  $$\rho_i = \text{tr}_{\{1, \ldots, n\}\setminus i}(\rho)$$
Quantum Multi-party Computing

- Players’ inputs form an arbitrary state $\rho$ in $H_1 \otimes H_2 \otimes \ldots \otimes H_n$
- Player $i$ holds $i$-th component:
  $\rho_i = \text{tr}_{\{1,\ldots,n\}\setminus i}(\rho)$
- Each player gets one output:
  $\rho_i' = \text{tr}_{\{1,\ldots,n\}\setminus i}(U\rho U^\dagger)$
Quantum Multi-party Computing

Even if $t$ out of $n$ players try to cheat:

1. Cheaters learn nothing (except output)
2. Cheaters cannot affect output (except by choice of inputs)
Easy Solution: Trusted Outside Mediator

• If everybody trusts Tom
• Send all inputs to Tom
• Tom:
  – Applies $U$
  – Distributes outputs
Easy Solution: Trusted Outside Mediator

• If everybody trusts Tom
• Send all inputs to Tom
• Tom:
  – Applies $U$
  – Distributes outputs

Challenge: Simulate the presence of Tom

$\rho' = U\rho U^\dagger$
Results

- $t < n/6$: Any Multi-party Quantum Computation

- $t < n/4$: Verifiable Secret-Sharing (weaker subtask)

- $t \geq n/4$: Even VQSS is impossible
Results

Quantum MPC

Verifiable Quantum Secret Sharing

Classical MPC (without broadcast)

Classical MPC (with broadcast)

? (Weaker task, to be defined)

IMPOSSIBLE

0 n/6 n/4 n/3 n/2

t = number of cheaters
MPQC and Fault-Tolerant Computing

• MPQC is like FTQC with a different error model...

<table>
<thead>
<tr>
<th></th>
<th>FTQC</th>
<th>MPQC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of errors</td>
<td>randomly spread, independent</td>
<td>maliciously placed, entangled with data</td>
</tr>
<tr>
<td>Error location</td>
<td>Can occur anywhere</td>
<td>At most $t$ positions</td>
</tr>
</tbody>
</table>

– Similar protocol techniques:

**Classical MPC** [BGW, CCD] $\rightarrow$ **FTQC** [AB99] $\rightarrow$ **MPQC** [us]

– Different proof techniques

(Need different notion of “proximity” to coding subspaces)
A Sketch of the Protocol
Protocol Overview

• **Share**
  – Each player encodes his input using a QECC
  – Sends $i$-th component to player $i$
  – Proves that sharing was done “correctly”
    i.e. distributed shares form a codeword except on positions held by cheaters

• **Compute**
  – Use fault-tolerant circuits to apply $U$ to encoded inputs

• **Distribute**
  – Give each player all components of his output
Why is this enough?

• **If:**
  - All players share their input with a “proper” codeword
  - (and) No information is leaked by proof

• **Then** the cheaters:
  - can’t **disturb** the calculation since QECC and FTQC will tolerate errors in any \( t \) locations
  - (Informally: ) can’t **learn info** since they can’t **disturb**!
An Impossibility Proof
Verifiable Quantum Secret-Sharing

• Idealized “qubit commitment”
• 2-phase protocol

• **Sharing**: Dealer $D$ shares a secret system $\rho$ such that
  – Cheaters can’t learn anything about $\rho$
  – Dealer can’t change $\rho$

• **Recovery**: Receiver $R$ specified by context
  – All players send shares to $R$
  – $R$ reconstructs $\rho$

**Easy Solution**: Give $\rho$ to trusted Tom, get it back later.
Verifiable Quantum Secret-Sharing

• Sharing phase of our **MPC** protocol is a **VQSS**

• **My opinion:**
  Most “interesting” **MPC** protocols will imply **VQSS**, since they should allow simulating Tom’s presence in more general tasks
  
  e.g. **qubit commitment**

• **Theorem:** **VQSS** is impossible for \( t \geq n/4 \)
Theorem: No VQSS tolerates $t \geq n/4$

Lemma:

Any VQSS protocol “is” a QECC correcting $t$ errors

Proof:

• Look at the state $F(|\psi\rangle)$ of protocol at the end of sharing phase when all players are honest, and input is $|\psi\rangle$

• Protocol is oblivious, so $F(|\psi\rangle) = E|\psi\rangle$ for some trace preserving $E$.

• At this point, arbitrary corruption of $t$ players can’t change reconstructed secret $|\psi\rangle$

• Thus $E$ is the encoding operator for a QECC.
Theorem: No VQSS tolerates $t \geq n/4$

Proof:

- No cloning says that no QECC can correct $n/2$ erasures
- Fact: Any QECC which corrects $t$ errors can correct $2t$ erasures
- Thus no QECC tolerates $n/4$ errors
- All these arguments work regardless of dimension of components of QECC
- Thus, no VQSS tolerates $t = n/4$ cheaters.
Conclusions

• Study general cryptographic tasks in distributed setting

• You can do anything you want when $t < n/6$

• You can’t do much when $t \geq n/4$

• Along the way:
  – First “zero-knowledge” quantum proofs secure against malicious verifiers
  – Refined notions of “proximity” to QECC’s.
  – Wrestled with definitions for malicious quantum adversaries
More Protocol Sketch
How to prove sharing is correct?

- Use Zero-Knowledge Proof techniques due to [Crépeau, Chaum, Damgård 1988] (from classical MPC)
- Based on classical Reed-Solomon code:
  - To encode $a$, pick a random polynomial $p$ of degree $2t$ over $\mathbb{Z}_q$ such that $p(0)=a$ and output $(p(1), \ldots, p(n))$
- We use: “polynomial codes” of [Aharonov, Ben-Or 1999]

$$E|a\rangle = \sum_{p: \text{deg}(p)=2t, p(0)=a} |p(1), p(2), \ldots, p(n)\rangle$$
Basic Step

• Prover takes secret $|\psi\rangle$
  – Shares $E|\psi\rangle$ (system #1)
  – Shares $E(\sum|a\rangle)$ (system #2)

• Players together generate random bit $b$

• If $b=0$ then do nothing
  If $b=1$ then “add in $Z_q$” System #1 to System #2

• Measure System #2 and broadcast results

• Accept if broadcast vector close to a classical codeword

$A(|x\rangle|y\rangle) = |x\rangle|y + x\rangle$

$A^{\otimes n}(E|\psi\rangle E\sum|a\rangle) = E|\psi\rangle E\sum|a\rangle$
Properties of Basic Step

- If dealer passes test many times in
  - computational basis and
  - Rotated “Fourier basis” (q-ary analogue of $|0\rangle + |1\rangle$, $|0\rangle - |1\rangle$)

  Then shared state is “close” to a quantum codeword

- If dealer was honest,

  then no information is leaked and state is not disturbed

- This can be “boosted” to get secure protocol for $t < n/4$
What does “close to a codeword” mean?

• Shared state should differ from a codeword only on positions held by cheaters

• Natural notion of closeness:

  (1) Reduced density matrix of honest players
      = reduced density matrix of some state in coding space $Q$

• Too strong: Our protocols can’t guarantee that.

• Instead:

  (2) Shares held by honest players pass parity checks restricted to those positions
What does “close to a codeword” mean?

- (1) $\neq$ (2)
  - (1) is not even a subspace!
  - Basic problem: errors and data can be entangled
- Analysis of fault-tolerant protocols only requires (1)
- We can only guarantee notion (2)
- Nonetheless, our protocols are secure:
  - Notion (2) strong enough to ensure well-defined decoding: changes made by cheaters to a state in (2) cannot affect output
  - Fault-tolerant procedures work for states in (2)