Quantum Computing as a Service

Secure and Verifiable Multi-Tenant Quantum Data Centre

Elham Kashefi
University of Edinburgh
Quantum Computing and Simulation Hub

CNRS Sorbonne University
Quantum Internet Alliance

VeriQloud
Currently
Quantum Links
Quantum Links

Unclonable / Measurement disturbance ... - security

QKD, Quantum Coin Flipping, ...
Quantum Links

Unclonable / Measurement disturbance … - security
QKD, Quantum Coin Flipping, …

Quantum Nodes
Quantum Links

Unclonable / Measurement disturbance … - security
QKD, Quantum Coin Flipping, …

Quantum Nodes

Superposition / Entanglement… - speed
Random Walk, Machine Learning, …
Quantum Links

Unclonable / Measurement disturbance… - security
QKD, Quantum Coin Flipping, …

Quantum Nodes

Superposition / Entanglement… - speed
Random Walk, Machine Learning, …
Future
Multi-Tenant Quantum Data Centre
Multi-Tenant Quantum Data Centre

Enhancing: efficiency, security, integrity
Use-Case Example: Privacy Preserving QML

Party with Data

Party with Q Algorithm

Party with Q Computer
Use-Case Example: Privacy Preserving QML

Quantum Secure Multi Party Computing
Plan
Plan

• 2 party QC: Honest Client - Malicious Server
Plan

• 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing
Plan

• 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing

• 2 party QC: Malicious Client - Malicious Server
Plan

• 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing

• 2 party QC: Malicious Client - Malicious Server
  - Quantum Cut and Choose Technique
Plan

• 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing

• 2 party QC: Malicious Client - Malicious Server
  - Quantum Cut and Choose Technique

• Multi party QC: Malicious Clients - Malicious Server
Plan

• 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing

• 2 party QC: Malicious Client - Malicious Server
  - Quantum Cut and Choose Technique

• Multi party QC: Malicious Clients - Malicious Server
  - Lifting Classical SMPC to QSMPC
Plan

- 2 party QC: Honest Client - Malicious Server
  - What is possible?
  - Building Blocks: QKD, Teleportation, Self-Testing
  - Verifiable Universal Blind Quantum Computing

- 2 party QC: Malicious Client - Malicious Server
  - Quantum Cut and Choose Technique

- Multi party QC: Malicious Clients - Malicious Server
  - Lifting Classical SMPC to QSMPC

- When can we have it for real?
Honest Client - Malicious Server

Party with Algorithm

Party with Data

Party with Q Computer

Clients → Internet → Server
Honest Client - Malicious Server

- Party with Algorithm
- Party with Data
- Party with Q Computer

Clients

Internet

Server

aws
Honest Client - Malicious Server

No privacy: Data, Algorithms, Results are all public
No Verification: The results are not classically simulatable
Secure Cloud Computing

Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?
Secure Cloud Computing

Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?

Limited Client  <->  Untrusted Server
Secure Cloud Computing

Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?

[Diagram showing a limited client communicating with an untrusted server]
Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?

**Secure Cloud Computing**
Secure Cloud Computing

Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?

Limited Client

Untrusted Server

X → Y
F(X) ← F(Y)

Y → F(Y)
Secure Cloud Computing

Rivest, Adleman and Dertouzos 1979
Can we process encrypted data without decrypting it first?

Gentry 2009 - Fully Homomorphic Encryption
computational security
Secure Classical access to Quantum Cloud?

Fillinger: No efficient informationally secure classical FHE scheme exist
Secure Classical access to Quantum Cloud?

Fillinger: No efficient informationally secure classical FHE scheme exist

Newman and Shi: No efficient informationally secure quantum FHE scheme exist
Secure Classical access to Quantum Cloud?

Fillinger: No efficient informationally secure classical FHE scheme exist

Newman and Shi: No efficient informationally secure quantum FHE scheme exist

Dunjko et.al.: No informationally secure quantum scheme for classical function evaluation (for restricted classical client)
Secure Classical access to Quantum Cloud?

Fillinger: No efficient informationally secure classical FHE scheme exist

Newman and Shi: No efficient informationally secure quantum FHE scheme exist

Dunjko et.al.: No informationally secure quantum scheme for classical function evaluation
(for restricted classical client)

Murimae: No informationally secure quantum scheme for quantum function evaluation
(for restricted classical client)
Secure Classical access to Quantum Cloud?
Secure Classical access to Quantum Cloud?

On the implausibility of informationally secure quantum cloud computing with Classical Client

*(PH collapses at the third level)*

Aaronson, Cojocaru, Gheorghiu, Kashefi, 2017
Generalised Encryption Scheme (GES)

[Abadi, Feigenbaum, Killian '87]

Client $x$ \vspace{0.4cm}
\hspace{1cm}$f(x)$ \vspace{0.4cm}
\hspace{2cm}BPP
\hspace{1cm}$x$\vspace{0.4cm}
\hspace{1cm}$y = Enc(x)$ \vspace{0.4cm}
\hspace{2cm}...\vspace{0.4cm}
\hspace{2cm}Server \vspace{0.4cm}
\hspace{2cm}|x| 

Information-theoretic security
Which functions admit a GES?

\[ \text{NP/poly} \cap \text{coNP/poly} \]

\[ \text{BPP} \]
What about **NP** functions?

Unless **PH** collapses
Generalised Encryption Scheme for QC (GES)

\[ f(x) \]

\[ y = Enc(x) \]

\[ f \in \text{BQP} \]

Information-theoretic security
1. Do $\text{BQP}$ functions admit a $\text{GES}$?

We give evidence that the answer is $\text{NO}$.
An oracle result

For each $d$, there exists an oracle, $O$, such that:

The oracle is based on Simon’s problem

$$O(n, x) = f_n(x)$$

Is $f_n$ 1-to-1 or does it have Simon’s property?

Simon’s property: $f_n$ is 2-to-1 and periodic
A sampling result

Unless, there exist circuits \( \{C_n\}_n \) having the properties:

\[
|C_n| = 2^n - \Omega(n/\log(n))
\]

\( C_n \) queries \( \text{NP}^{\text{NP}} \)

Computes exactly the permanent of \( n \times n \) matrix

Best known algorithm for permanent (Ryser ’63): \( O(n2^n) \)
A sampling result

SampBQP  SampBPP  SampGES

Unless, there exist circuits having the properties:

\[ C_n \mid C_n \mid = 2^n \cap (n / \log(n)) \]

\[ C_n \text{ queries} \]

NP

Computes exactly the permanent of \( n \times n \) matrix

Best known algorithm for permanent (Ryser '63): \( O(n2^n) \)

GES for SampBQP \( \rightarrow \) “efficient” circuits for permanent
Secure Classical Access to Quantum Cloud

Limited Client

Untrusted Server

\[ F(X) \rightarrow \text{NOT POSSIBLE} \]

\[ F(Y) \]
Secure Quantum access to Quantum Cloud
Secure Quantum access to Quantum Cloud

Limited QClient <-> Untrusted Server
Secure Quantum access to Quantum Cloud

Limited QClient

Untrusted Server
Secure Quantum access to Quantum Cloud

Limited QClient \[|X\rangle\] \[|Y\rangle\] \[\rightarrow\] Quantum Links \[\leftarrow\] Untrusted Server
Secure Quantum access to Quantum Cloud

Limited QClient \[|X\rangle\] → \[|Y\rangle\] Quantum Links

Untrusted Server \[|Y\rangle\] → \[U|Y\rangle\]
Secure Quantum access to Quantum Cloud
Secure Quantum access to Quantum Cloud

Limited QClient → Quantum Links → Untrusted Server

Broadbent, Fitzsimons, and Kashefi 2009 - Universal Blind Quantum Computing
Informational security
Secure Quantum access to Quantum Cloud

QKD for encoding

Limited QClient

Quantum Links

Untrusted Server

|X> → |Y>

U|X> → U|Y>

|Y> → U|Y>

Broadbent, Fitzsimons, and Kashefi 2009 - Universal Blind Quantum Computing
Informational security
Secure Quantum access to Quantum Cloud

QKD for encoding

Teleportation for computing

Quantum Links

Limited QClient

Untrusted Server

Broadbent, Fitzsimons, and Kashefi 2009 - Universal Blind Quantum Computing
Informational security
Secure Quantum access to Quantum Cloud

- **QKD** for encoding
- **Teleportation** for computing
- **Limited QClient**
- **Untrusted Server**
- **Testing** for verification

Broadbent, Fitzsimons, and Kashefi 2009 - Universal Blind Quantum Computing

Informational security
Computing with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]
Computing with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

gate teleportation

\[ |\phi\rangle \quad \xrightarrow{\pm_{\alpha}} \quad J(\alpha)(|\phi\rangle) \]

\[ |+\rangle \quad \xrightarrow{X} \quad J(\alpha)(|\phi\rangle) \]
Computing with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

gate teleportation

\[ |\phi\rangle \rightarrow |+\rangle \rightarrow J(\alpha)(|\phi\rangle) \]
Computing with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

gate teleportation

\[ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{i\alpha} \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -e^{i\alpha} \end{pmatrix} \]

|\phi\rangle \rightarrow \pm_\alpha \rightarrow X \rightarrow J(\alpha)(|\phi\rangle)
Computing with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

gate teleportation

\[ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ e^{i\alpha} \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -e^{i\alpha} \end{pmatrix} \]

\[ \begin{pmatrix} 0 & 1 \\ X \end{pmatrix} \]

\[ J(\alpha)(|\phi\rangle) \]
Hiding with Teleportation

$$J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix}$$

Single qubit rotation

Quantum Computer

$$|\phi\rangle$$

$$|+\rangle$$

$$J(\alpha)(|\phi\rangle)$$
Hiding with Teleportation

\[
J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix}
\]

Single qubit rotation $Z(\theta) |\phi\rangle$ followed by $|\pm_{\alpha+\theta}\rangle$ and then $J(\alpha)(|\phi\rangle)$.
Hiding with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

Single qubit rotation

\[ Z(\theta) |\phi\rangle \rightarrow |\pm_{\alpha+\theta}\rangle \]

Quantum Computer

\[ J(\alpha + \theta) \]

\[ J(\alpha)(|\phi\rangle) \]
Hiding with Teleportation

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]

Single qubit rotation

\[ Z(\theta) |\phi\rangle \rightarrow |\pm_{\alpha+\theta}\rangle \]

\[ J(\alpha + \theta) \]

Quantum Computer

\[ J(\alpha) (|\phi\rangle) \]

Hiding the Angles
Hiding the measurement result

\[ J(\alpha) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix} \]
Gates Composition

\[|+\theta\rangle \rightarrow \delta \rightarrow X \rightarrow \delta \rightarrow X \rightarrow \delta \rightarrow X \rightarrow \oplus r \rightarrow |+\rangle \]

\[|+\rangle \rightarrow \Delta \rightarrow X \rightarrow \Delta \rightarrow X \rightarrow \Delta \rightarrow X \rightarrow \oplus r \rightarrow |+\rangle \]

\[\text{Local Cli} \quad X \rightarrow \Delta \rightarrow X \rightarrow \oplus r \rightarrow |+\rangle \]
Gates Composition
Gates Composition

Perfect decryption and encryption at each step

Client-Server interactions
Re-writing

\[ |+\theta\rangle_1 \]
\[ |+\theta\rangle_2 \]

\[ \delta_1 \]
\[ \delta_2 \]

\[ Z \]
\[ X \]

Out

Out
Re-writing

\[ |+\theta_1\rangle \]
\[ |+\theta_2\rangle \]
\[ \delta_1 \]
\[ \delta_2 \]

\[ Z \quad \text{Out} \]
\[ X \quad \text{Out} \]
Universal Blind Quantum Computings

\[ X = (\tilde{U}, \{ \phi_{x,y} \}) \]
Universal Blind Quantum Computings

$X = (\tilde{U}, \{\phi_{x,y}\})$

random single qubit generator

$\frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle)$

$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$
Universal Blind Quantum Computing

$X = (\tilde{U}, \{\phi_{x,y}\})$

random single qubit generator

$\frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle)$

$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$
Universal Blind Quantum Computings

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]

\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]
Universal Blind Quantum Computing

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]
\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]

\[ r_{x,y} \in \mathbb{R} \{0, 1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]
Universal Blind Quantum Computings

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]
\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]

\[ r_{x,y} \in R \{0, 1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]
Universal Blind Quantum Computings

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]

\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]

\[ r_{x,y} \in R \{0, 1\} \]

\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]

\[ \{ |+\delta_{x,y}\rangle, |-\delta_{x,y}\rangle \} \]
Universal Blind Quantum Computing

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]
\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]

\[ r_{x,y} \in \mathbb{R} \{0, 1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]

\[ s_{x,y} \in \{0, 1\} \]
\[ \{ |+\delta_{x,y}\rangle, |-\delta_{x,y}\rangle \} \]
Universal Blind Quantum Computings

\[ X = (\tilde{U}, \{\phi_{x,y}\}) \]

random single qubit generator

\[ \frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle) \]
\[ \theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4 \]

\[ r_{x,y} \in R \{0,1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]

\[ s_{x,y} = s_{x,y} + r_{x,y} \]
\[ s_{x,y} \in \{0,1\} \]
\[ \{ |+\delta_{x,y}\rangle, |-\delta_{x,y}\rangle \} \]
**Security Definition**

Protocol $P$ on input $X = (\tilde{U}, \{\phi_{x,y}\})$ leaks at most $L(X)$

- The distribution of the classical information obtained by Server is independent of $X$
- Given the above distribution, the quantum state is fixed and independent of $X$
What about correctness?
What about correctness?

- **Correctness**: in the absence of any deviation, client accepts and the output is correct.

- **Soundness**: Client rejects an incorrect output, except with probability at most exponentially small in the security parameter.
Verification of Quantum Computing

Self Testing 2005
Decide if the physical devices simulate their specification
Verification of Quantum Computing

Single-prover prepare-and-send

verifier has the ability to prepare quantum states and send them to the prover

- State authentication-based protocols
- Trapification-based protocols
- Test or Compute

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Verifier resources</th>
<th>Communication</th>
<th>2-way quantum comm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clifford-QAS VQC</td>
<td>$O(\log(1/\epsilon))$</td>
<td>$O(N \cdot \log(1/\epsilon))$</td>
<td>Y</td>
</tr>
<tr>
<td>Poly-QAS VQC</td>
<td>$O(\log(1/\epsilon))$</td>
<td>$O((n + L) \cdot \log(1/\epsilon))$</td>
<td>N</td>
</tr>
<tr>
<td>VUBQC</td>
<td>$O(1)$</td>
<td>$O(N \cdot \log(1/\epsilon))$</td>
<td>N</td>
</tr>
<tr>
<td>Test-or-Compute</td>
<td>$O(1)$</td>
<td>$O((n + T) \cdot \log(1/\epsilon))$</td>
<td>N</td>
</tr>
</tbody>
</table>
Verification of Quantum Computing

Single-prover receive-and-measure

verifier receives quantum states from the prover and has the ability to measure them

- Post-hoc Verification (none hiding)
- Measuring only blind QC

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Measurements</th>
<th>Observables</th>
<th>Blind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement-only</td>
<td>$O(N \cdot 1/\alpha \cdot 1/\epsilon^2)$</td>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>Hypergraph measurement-only</td>
<td>$O(\max(N, 1/\epsilon^2)^{22})$</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>1S-Post-hoc</td>
<td>$O(N^2 \cdot \log(1/\epsilon))$</td>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>Steering-based VUBQC</td>
<td>$O(N^{13} \cdot \log(N) \cdot \log(1/\epsilon))$</td>
<td>5</td>
<td>Y</td>
</tr>
</tbody>
</table>
Multi-prover entanglement-based

Classical Verifier interacts with more than one provers that are not allowed to communicate during the protocol

- CHSH game Rigidity
- Self-testing graph states
- Pauli Braiding

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Provers</th>
<th>Qmem provers</th>
<th>Rounds</th>
<th>Communication</th>
<th>Blind</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUVD</td>
<td>2</td>
<td>2</td>
<td>$O(N^{8192} \cdot \log(1/\epsilon))$</td>
<td>$O(N^{8192} \cdot \log(1/\epsilon))$</td>
<td>Y</td>
</tr>
<tr>
<td>McKague</td>
<td>$O(N^{22} \cdot \log(1/\epsilon))$</td>
<td>0</td>
<td>$O(N^{22} \cdot \log(1/\epsilon))$</td>
<td>$O(N^{22} \cdot \log(1/\epsilon))$</td>
<td>Y</td>
</tr>
<tr>
<td>GKW</td>
<td>2</td>
<td>1</td>
<td>$O(N^{2048} \cdot \log(1/\epsilon))$</td>
<td>$O(N^{2048} \cdot \log(1/\epsilon))$</td>
<td>Y</td>
</tr>
<tr>
<td>HPDF</td>
<td>$O(N^4 \log(N) \cdot \log(1/\epsilon))$</td>
<td>$O(\log(1/\epsilon))$</td>
<td>$O(N^4 \log(N) \cdot \log(1/\epsilon))$</td>
<td>$O(N^4 \log(N) \cdot \log(1/\epsilon))$</td>
<td>Y</td>
</tr>
<tr>
<td>FH</td>
<td>5</td>
<td>5</td>
<td>$O(N^{16} \cdot \log(1/\epsilon))$</td>
<td>$O(N^{19} \cdot \log(1/\epsilon))$</td>
<td>N</td>
</tr>
<tr>
<td>NV</td>
<td>7</td>
<td>7</td>
<td>$O(1)$</td>
<td>$O(N^3 \cdot \log(1/\epsilon))$</td>
<td>N</td>
</tr>
</tbody>
</table>
Verification of Quantum Computing

- Overhead
- Noise
- Scalability
Unconditionally Verifiable Blind Quantum Computing

Fitzsioms Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsioms Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
Trapification

Unconditionally Verifiable Blind Quantum Computing

Fitzsimons Kashefi, 2012
\[ \Omega_{Eve,system} \]
Trapification

\[ \Omega_{Eve,system} \]

Security

\[ \sigma_{testsubspace} \]
Trapification

\[ \Omega_{Eve,system} \]

Security

\[ \sigma_{testsubspace} \]

Trap Measurements

\[
M^\theta |+\theta\rangle \rightarrow s = 0 \\
M^\theta |-\theta\rangle \rightarrow s = 1
\]
Trapification

Ω_{Eve, system}

Security

σ_{testsubspace}

Trap Measurements

M^θ |+θ⟩ → s = 0
M^θ |−θ⟩ → s = 1

Prob trap being correct and the computation is wrong is bounded
\[ \Omega_{Eve,system} \]

\[ \sigma_{testsubspace} \]

**Trap Measurements**

\[ M^\theta |+\theta\rangle \rightarrow s = 0 \]

\[ M^\theta |-\theta\rangle \rightarrow s = 1 \]

**Prob trap being correct and the computation is wrong is bounded**

\[ \sum_\nu p(\nu) \quad \text{Tr} \left( P^\nu_{\text{incorrect}} B(\nu) \right) \leq \epsilon \]

\[ P^\nu_{\text{incorrect}} := P_\perp \otimes |acc\rangle\langle acc| \]
Robust Verifiable Secure Quantum Access to Noisy Quantum Qloud

Classical input/output

Perfect blindness and exponential verification

Exponential correctness on honest-but-noisy device

No overhead besides repetitions

Securing Quantum Computations in the NISQ Era

Kashefi, Leichtle, Music, Ollivier, 2020
Robust Verifiable Secure Quantum Access to Noisy Quantum Qloud

- Classical input/output
- Perfect blindness and exponential verification
- Exponential correctness on honest-but-noisy device
- No overhead besides repetitions

Securing Quantum Computations in the NISQ Era

Kashefi, Leichtle, Music, Ollivier, 2020
Robust Verifiable Secure Quantum Access to Noisy Quantum Qloud

Classical input/output

Perfect blindness and exponential verification

Exponential correctness on honest-but-noisy device

No overhead besides repetitions

Securing Quantum Computations in the NISQ Era

Kashefi, Leichtle, Music, Ollivier, 2020
Secure Classical Access to Quantum Cloud

Limited Client

\( F(X) \)

\( X \) → \( F(Y) \)

Untrusted Server

\( F(Y) \)

\( \text{NOT POSSIBLE} \)
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?

Classical Client Quantum FHE
Mahadev, FOCS 2018
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?

Classical Client Quantum FHE
Mahadev, FOCS 2018

Delegated Pseudo-Secret Random Qubit Generator
Cojocaru, Colisson, Kashefi, Wallden, AsiaCrypt 2019
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?

Classical Client Quantum FHE
Mahadev, FOCS 2018

Delegated Pseudo-Secret Random Qubit Generator
Cojocaru, Colisson, Kashefi, Wallden, AsiaCrypt 2019

\[ X = (\bar{U}, \{\phi_{x,y}\}) \]

\[ r_{x,y} \in_R \{0, 1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]

\[ s_{x,y} := s_{x,y} + r_{x,y} \]
\[ s_{x,y} \in \{0, 1\} \]
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?

Classical Client Quantum FHE
Mahadev, FOCS 2018

Delegated Pseudo-Secret Random Qubit Generator
Cojocaru, Colisson, Kashefi, Wallden, AsiaCrypt 2019

\[ X = (\bar{U}, \{\phi_{x,y}\}) \]

\[ r_{x,y} \in_R \{0, 1\} \]
\[ \delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y} \]
\[ s_{x,y} := s_{x,y} + r_{x,y} \]
\[ s_{x,y} \in \{0, 1\} \]
Computationally Secure (Post-quantum safe) Classical Access to Quantum Cloud?

Classical Client Quantum FHE
Mahadev, FOCS 2018

Delegated Pseudo-Secret Random Qubit Generator
Ciofdu, Collison, Kashefi, Wallden, AsiaCrypt 2019

$X = \{\bar{U}, \{\phi_{x,y}\}\}$

$O(1000)$ server qubits for randomising one single client qubit

$r_{x,y} \in_R \{0,1\}$

$\delta_{x,y} = \phi_{x,y} + \theta_{x,y} + \pi r_{x,y}$

$s_{x,y} \in \{0,1\}$

$\{+\delta_{x,y}, -\delta_{x,y}\}$
Secure Access to Quantum Cloud

\[ \equiv \]

Quantum Communication
Malicious Client - Malicious Server

Party with Q Computer

Party with Algorithm

Party with Data

Clients

Q Internet

Server
Yao Garbled Circuit - Secure 2-party Computing

Secret input $a$

Garbled Program $f$

Yao 1986
Yao Garbled Circuit - Secure 2-party Computing

\[ A_0, A_1 \]
\[ B_0, B_1 \]
\[ C_0, C_1 \]
\[ D_0, D_1 \]
\[ E_0, E_1 \]
\[ G_0, G_1 \]
\[ H_0, H_1 \]
\[ F_0, F_1 \]
\[ I_0, I_1 \]

\[ \text{Enc}_{C_0,D_0}(F_1) \]
\[ \text{Enc}_{C_0,D_1}(F_1) \]
\[ \text{Enc}_{C_1,D_0}(F_1) \]
\[ \text{Enc}_{C_1,D_1}(F_0) \]
\[ \text{Enc}_{A_0,B_0}(E_0) \]
\[ \text{Enc}_{A_0,B_1}(E_1) \]
\[ \text{Enc}_{E_0,F_0}(G_0) \]
\[ \text{Enc}_{E_0,F_1}(G_1) \]
\[ \text{Enc}_{E_0,G_0}(H_0) \]
\[ \text{Enc}_{E_0,G_1}(H_0) \]
\[ \text{Enc}_{E_1,F_0}(G_0) \]
\[ \text{Enc}_{E_1,F_1}(G_1) \]
\[ \text{Enc}_{E_1,G_0}(H_0) \]
\[ \text{Enc}_{E_1,G_1}(H_1) \]
\[ \text{Enc}_{G_0,F_0}(I_0) \]
\[ \text{Enc}_{G_0,F_1}(I_1) \]
\[ \text{Enc}_{G_1,F_0}(I_1) \]
\[ \text{Enc}_{G_1,F_1}(I_0) \]

Yao 1986
Yao Garbled Circuit - Secure 2-party Computing

Garbled Circuit

Secret input \(a\)
Yao Garbled Circuit - Secure 2-party Computing

Secret input $a$

Garbled Program $f$
Yao Garbled Circuit - Secure 2-party Computing

Secret input a

Garbled Program f

Insert secret input b
Evaluate f(a,b)

Yao 1986
Yao Garbled Circuit - Secure 2-party Computing

- Secret input a
- Garbled Program f
- Computational Security
  - Requires OT
- Honest but Curious Adversary

Yao 1986
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map

$\theta \ldots \theta'$

$\frac{1}{\sqrt{2}} (|0\rangle + e^{i\theta} |1\rangle)$

$|0\rangle, |1\rangle$
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map

Kashefi, Walden 16
Kashefi, Music, Wallden 17
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map

Server’s input placed in DT(G) with corresponding trap-colouring

Verifiable Quantum Yao

Kashefi, Walden 16
Kashefi, Music, Wallden 17
Verifiable Quantum Yao

Secret input q_c

Garbled CP map

---

Fig. 2. Server gives their input (blue) and client chooses (randomly) where in the input base-location to place the input. The random choice is highlighted. The trap-colouring is filled correspondingly, after the random choice is made.

Finally, after the server has received all the qubits, announces the secret keys \((m_{x,i}, m_{z,i})\) for each input \(i\) to the client, so that the client can update the encryption for these qubits and have \((x_i, \chi_i) = (x_0 + m_{x,i}, \pi m_{z,i})\).

With the updated encryption, the client computes the suitable measurement angles \(\theta_i\). It is worth pointing out that the key releasing step from server to client could be avoided, by using classical OT to compute the measurement angles as a function of the secret parameters of the server \((m_{x,i}, m_{z,i})\) for the first two layers (that have dependency on \(m_{x,i}, m_{z,i}\)). While this could be necessary for future work, to construct protocols dealing with malicious client, it is not necessary for our case where the client is considered to be specious.

3.2 Server’s output extraction

In the regular VUBQC protocol, the server returns all the output qubits to the client. The client measures the final layer’s traps to check for any deviation and then obtains the output of the computation by decrypting the output computation qubits using their secret keys. In the 2PQC, part of the output (of known base-locations) should remain in the hands of the server. This, however, would not allow the client to check for the related traps (that could have effects on other output qubits). Similar to the input injection, the solution is obtained via an extra layer of encryption by server followed by a delayed key releasing.
Verifiable Quantum Yao

Secret input q_c

Garbled CP map

\[ r_{x,y} \in_R \{0,1\} \]
\[ \delta_{x,y} = \phi_{x,y} + \theta_{x,y} + \pi r_{x,y} \]
Verifiable Quantum Yao

Secret input q_c

Garbled CP map

Insert secret input q_c

Evaluate CP(q_c,q_s)
Verifiable Quantum Yao

Secret input $q_c$

Garbled CP map

Informationally Secure

*(needs classical SMPC for angle evaluations)*

Quantum Honest but Curious Client

Requires classical $O(N)$ online communication

Insert secret input $q_c$

Evaluate CP($q_c, q_s$)

Kashefi, Walden 16
Kashefi, Music, Wallden 17
Boosting Security
(Semi-Malicious Client to Fully Malicious one)

Cut: Sender sends multiple copies of a state and message (with independent randomness) to the Receiver.

Choose: Server chooses 1 copy for executing the rest of the protocol (evaluation state) and checks that the s-1 other (check states) were correctly constructed by asking the Sender to send proofs and measuring them accordingly.

Conditions for applying Q-CC

✓ Practical Efficient Malicious Client - Malicious Server

States) where correctly constructed by asking the Sender to send proofs and measuring them accordingly.

Client manipulates single qubit

Kashefi, Music, Unruh, Wallden 2021
Malicious Clients - Malicious Server

Party with Algorithm

Party with Data

Q Internet

Clients

Server
Garbled her part of the CP map

\[ \theta \]

Secret input \( q_1 \)

Garbled her part of the CP map

\[ \theta' \]

Secret input \( q_n \)
Multiparty Delegated Quantum Computing 2017

Garbled her part of the CP map

Secret input $q_1$

Secret input $q_n$

Kashefi, Pappa 2017
Garbled her part of the CP map

Secret input $q_1$

Secret input $q_n$

Garbled her part of the CP map
Secret input $q_1$

Garbled her part of the CP map

Secret input $q_n$

Garbled her part of the CP map

$$\theta_j = \theta_j^i + \sum_{k=1, k \neq j}^{n} (-1)^{\bigoplus_{i=k}^{n} t_j^i} \theta_k^j$$

Kashefi, Pappa 2017
Secret input q_1
Garbled her part of the CP map

Secret input q_n
Garbled her part of the CP map

Kashefi, Pappa 2017
Multiparty Delegated Quantum Computing 2017

Secret input q₁
Garbled her part of the CP map

. . .

Secret input qₙ
Garbled her part of the CP map

. . .

Secret input q₁
Garbled her part of the CP map

. . .

Secret input qₙ
Garbled her part of the CP map

Kashefi, Pappa 2017
Garbled her part of the CP map

\[ \delta_j = \phi_j' + \pi \bigoplus_{k=1}^{n} r_j^k + \theta_j \]

Secret input q_1

Garbled her part of the CP map

Secret input q_n

Garbled her part of the CP map

Theorem 1

Theorem 2

Secret input q_1

Garbled her part of the CP map

Secret input q_n

Garbled her part of the CP map

Kashefi, Pappa 2017
Garbled her part of the CP map

Secret input q_1

Garbled her part of the CP map

Secret input q_n

Informationally Secure
Classical SMPC is needed
No client-server colluding is allowed!
Multiparty Delegated Quantum Computing 2017
Clients can insert traps only in their subgraphs

**But**

A connected path for computation can be obtained only if they collaborate

**But**

They need not to leak the position of traps
Clients can insert traps only in their subgraphs

But

A connected path for computation can be obtained only if they collaborate

But

They need not to leak the position of traps

In Symmetric Case these issues are resolved by Dulek, Grilo, Jeffery, Majenz, Schaffner 2020
Multiparty Delegated Quantum Computing 2021

Double Blind QC - a classically orchestrated delegation

Good Enough State - correct up to a deviation independent of the inputs and security parameters
Steps to be updated to transform into a multi-client setting &
Conditions that these replacement need to satisfy
VUBQC Deconstruction - Reconstruction

Steps to be updated to transform into a multi-client setting &
Conditions that these replacement need to satisfy

[Diagram of client-server interaction]
VUBQC Deconstruction - Reconstruction

Steps to be updated to transform into a multi-client setting & Conditions that these replacement need to satisfy
VUBQC Deconstruction - Reconstruction

Steps to be updated to transform into a multi-client setting &
Conditions that these replacement need to satisfy

Collaboratively prepared

Collaboratively measured
Replacing Classical Steps with Classical SMPC
Replacing Classical Steps with Classical SMPC

Possibly deviated multi party encrypted state (independent of secret parameters)
Double Blind QC

Computation:

\[ M(\delta_1), M(\delta_2), M(\delta_3) \]

Constraints:

\[ b_1 = 1, b_2 = 0, b_3 = 1 \]
Double Blind QC

Classical SMPC

Verifier

Prover

$M(\delta_1), M(\delta_2), M(\delta_3)$

$b_1 = 1, b_2 = 0, b_3 = 1$
Double Blind QC

Classical SMPC

Realised itself by a UBQC pattern
Double Blind Gadgets for $H$ or $I$
Double Blind Gadgets for $H$ or $I$

Clients: sends encrypted input and rotated states

SMPC: redistribute them to become dummy or trap
## Multiparty Delegated Quantum Computing 2021

**Dulek, Grilo, Jeffery, Majenz, Schaffner 2020**

**Alon, Chung, Chung, Huang, Lee, Shen**

<table>
<thead>
<tr>
<th>Metric</th>
<th><strong>[9]</strong></th>
<th><strong>[26]</strong></th>
<th><strong>[1]</strong></th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Stat. upgrade of CSMPC</td>
<td>Statistical</td>
<td>Comp. (FHE + CSMPC)</td>
<td>Stat. upgrade of CSMPC</td>
</tr>
<tr>
<td>Abort</td>
<td>Unanimous</td>
<td>Unanimous</td>
<td>Identifiable</td>
<td>Unanimous</td>
</tr>
<tr>
<td>Composability</td>
<td>Composable</td>
<td>Stand-Alone</td>
<td>Stand-Alone</td>
<td>Composable</td>
</tr>
<tr>
<td>Max Malicious Players</td>
<td>$N - 1$</td>
<td>$\left\lceil \frac{C_{\text{dist}} - 1}{2} \right\rceil$</td>
<td>$N - 1$</td>
<td>$N - 1$</td>
</tr>
<tr>
<td>Protocol Nature</td>
<td>Symmetric</td>
<td>Symmetric</td>
<td>Semi-Delegated</td>
<td>Delegated</td>
</tr>
<tr>
<td>Network Topology</td>
<td>Q and C: Complete</td>
<td>Q and C: Complete</td>
<td>Q and C: Complete</td>
<td>Q: Star / C: Complete</td>
</tr>
<tr>
<td>Q Operations</td>
<td>F.T. Q. Comp</td>
<td>FT Q Comp</td>
<td>FT Q Comp</td>
<td>Cl.: Single Qubit</td>
</tr>
<tr>
<td>Classical SMPC</td>
<td>Clifford Computation, Operations in $\mathbb{Z}_2$, CT $\mathcal{O}(g + \eta(N + t))$</td>
<td>CT</td>
<td>Clifford Computation, FHE verification $\mathcal{O}(1)$</td>
<td>Serv.: FT Q Comp</td>
</tr>
<tr>
<td>Rounds (C or CSMPC)</td>
<td></td>
<td></td>
<td></td>
<td>Operations in $\mathbb{Z}_8$, $\mathbb{Z}_2$, CT $\mathcal{O}(d + 5)$</td>
</tr>
<tr>
<td>Rounds (Q)</td>
<td>Par.: $\mathcal{O}(Nd)$</td>
<td>Par.: 3 (if C output)</td>
<td>Par.: $\mathcal{O}(N^4)$</td>
<td>Par.: 2 (1 if C output)</td>
</tr>
<tr>
<td></td>
<td>Seq.: $\mathcal{O}(N(N + t + c))$</td>
<td>Seq.: $\mathcal{O}(\eta^2(N + t))$</td>
<td>Seq.: $\mathcal{O}(\eta N d)$</td>
<td>Serv. (par.): $\mathcal{O}(\eta N^2 d)$</td>
</tr>
<tr>
<td>Size of Q Memory</td>
<td>Par.: $\mathcal{O}(\eta^2(N + t))$</td>
<td>Par.: $\mathcal{O}(\eta^2 N(N + t))$</td>
<td>Par.: $\mathcal{O}(t N^9 \eta^2)$</td>
<td>Cl.: 3 (0 if C I&amp;O)</td>
</tr>
<tr>
<td></td>
<td>Seq.: $\mathcal{O}(\eta^2 N)$</td>
<td>Seq.: $\mathcal{O}(N^2)$</td>
<td>Serv. (seq.): $\mathcal{O}(\eta N d)$</td>
<td>Serv. (seq.): $\mathcal{O}(\eta N d)$</td>
</tr>
</tbody>
</table>

**Lipinska, Ribeiro, Wehner 2020**
Practical Efficient Malicious Clients - Malicious Server?
Practical Efficient Malicious Clients - Malicious Server?

Each Module Can be Optimised

- SMPC: angles evaluations and permutations
- Remote State Prep: Hardware Dependent
- Blind QC: Not every qubits being hidden
- Verifiable QC: No Need for dummies
Key component - Remote State Preparation
Key component - Remote State Preparation

\[ |+\theta^1_j\rangle \rightarrow \text{RSP} \rightarrow \begin{cases} |+\theta\rangle & \text{or abort} \\ \\
|+\theta^2_j\rangle \\
|+\theta^3_j\rangle \\
\vdots \\
|+\theta^n_j\rangle \end{cases} \]

\[ \theta_j = \theta^j_j + \sum_{k=1, k \neq j}^{\infty} (-1)^{i_k} t^i_j + a_j \theta^k_j \]
The Most Optimal Client-Server RSP

Quantum Enclave - Remote State Rotation

\[ (s, \rho_{out}) = (\theta, Z(\theta)\rho_{in}Z^*(\theta)) \]

Arapinis, Chakraborty, Kaplan, Kashefi, Ma, 2021
The Most Optimal Multi Party QSMPC

Qline Architecture + Remote State Rotation + QSMPC

VeriQloud’s fully connected quantum network with a single optical fibre
A Secure New World