

# Introduction to Quantum Computing

## Lecture 19: Universal Blind Quantum Computing

Petros Wallden

School of Informatics, University of Edinburgh

28th November 2023



- 1 Blind Quantum Computing: What & Why
- 2 Tools for MBQC-based Universal Blind Quantum Computing
- 3 UBQC protocol and Verification

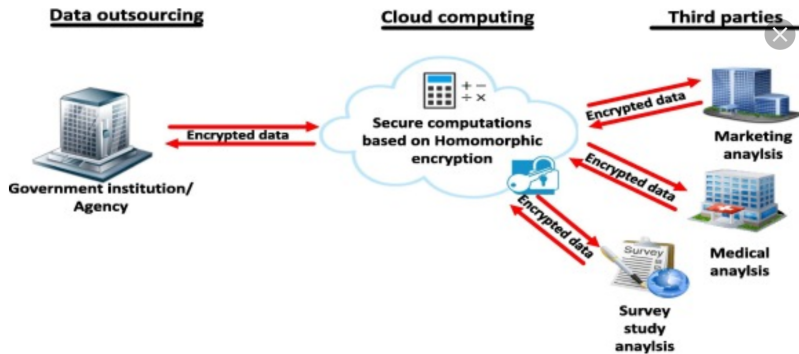
## Blind Quantum Computing: What & Why

# Secure Cloud Computing

## Modern Cyber Security goes beyond encryption

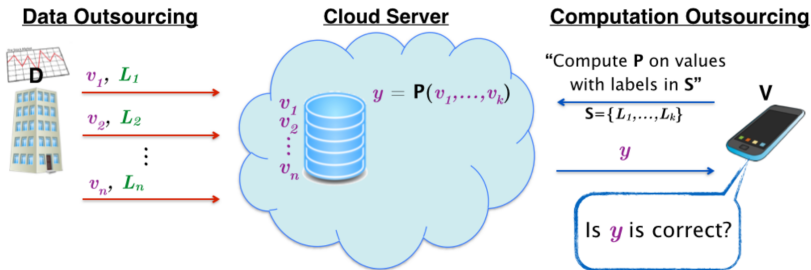
(e.g. Privacy-preserving Data Mining)

Delegated Private Computation (e.g. sensitive medical data)



## Modern Cyber Security goes beyond encryption (e.g. Privacy-preserving Data Mining)

### Verified Delegated Private Computation

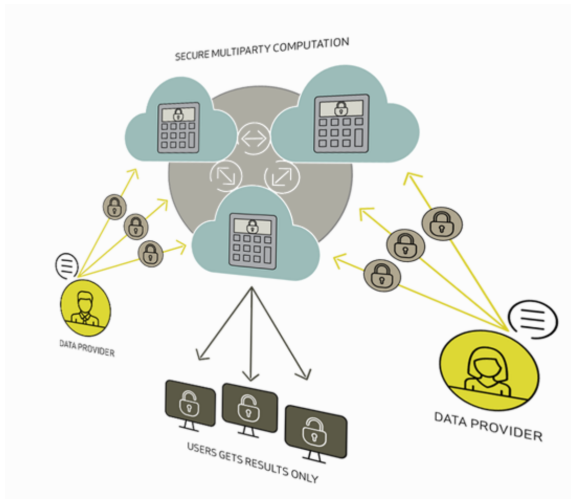


# Secure Cloud Computing

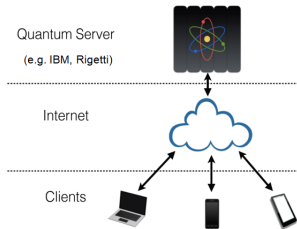
## Modern Cyber Security goes beyond encryption

(e.g. Privacy-preserving Data Mining)

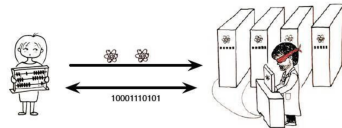
Secure Multiparty Computation (e.g. e-voting, auctions)



# The Secure Quantum Cloud



- Clients want to maintain **privacy, accuracy** and **reliability**
- Clients want to use the **extra power of quantum computing**



- Universal Blind Quantum Computation (Broadbent, Fitzsimons, Kashefi 2009)
- Basis for numerous extra functionalities
- Client sends **random single qubits** to Server

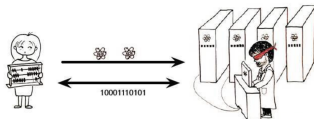
# The Secure Quantum Cloud

- Realistic setting (few large quantum computers)
- Active area to obtain efficient quantum analogues (e.g.):



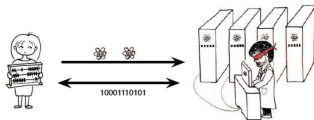
# The Secure Quantum Cloud

- Realistic setting (few large quantum computers)
- Active area to obtain efficient quantum analogues (e.g.):  
**Secure Quantum Cloud**



# The Secure Quantum Cloud

- Realistic setting (few large quantum computers)
- Active area to obtain efficient quantum analogues (e.g.):  
**Secure Quantum Cloud**

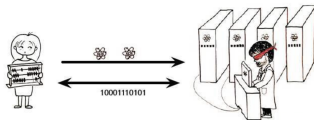


## Verifiable Secure Quantum Cloud



# The Secure Quantum Cloud

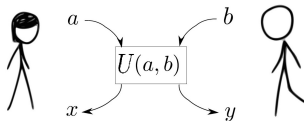
- Realistic setting (few large quantum computers)
- Active area to obtain efficient quantum analogues (e.g.):  
**Secure Quantum Cloud**



## Verifiable Secure Quantum Cloud



## Secure Two-Party Quantum Computation



## Tools for MBQC-based Universal Blind Quantum Computing

# Blind Computation Setting

## **Alice/Client:**

- Limited computational power
- Wants to use a quantum computer
- Does not trust Bob/Server

## **Alice/Client:**

- Limited computational power
- Wants to use a quantum computer
- Does not trust Bob/Server

## **Bob/Server:**

- Has universal quantum computer
- Is willing to help Alice
- Will not lend Alice his device

# Blind Computation Setting

## Alice/Client:

- Limited computational power
- Wants to use a quantum computer
- Does not trust Bob/Server

## Bob/Server:

- Has universal quantum computer
- Is willing to help Alice
- Will not lend Alice his device

## Blind Computation $\Rightarrow$ Bob cannot determine Alice's:

- Input
- Intended output
- Computation (not required for fully homomorphic encryption)

Alice must encrypt everything (input, computation, output)

- **Use of MBQC** (possible otherwise)
  - **Alice's power:**
    - 1 **Can** prepare single qubits
    - 2 **Cannot** measure, store, prepare entangled qubits, apply unitary gates



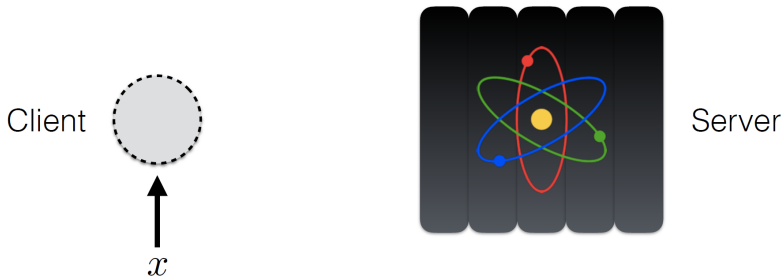
- **Use of MBQC** (possible otherwise)
  - **Alice's power:**
    - 1 **Can** prepare single qubits
    - 2 **Cannot** measure, store, prepare entangled qubits, apply unitary gates
  - **Alice:**
    - 1 Sends single qubits to Bob (**not** of the  $|+\rangle$  form)
    - 2 Instructs Bob to **entangle** them ( $\wedge Z$ ) as in a cluster state for MBQC and then **measure** them after further interaction

- **Use of MBQC** (possible otherwise)
  - **Alice's power:**
    - 1 **Can** prepare single qubits
    - 2 **Cannot** measure, store, prepare entangled qubits, apply unitary gates
  - **Alice:**
    - 1 Sends single qubits to Bob (**not** of the  $|+\rangle$  form)
    - 2 Instructs Bob to **entangle** them ( $\wedge Z$ ) as in a cluster state for MBQC and then **measure** them after further interaction
  - **Bob:**
    - 1 Does **not** know what states Alice sends him
    - 2 Follows instructions; returns measurement outcomes to Alice

**Blindness** w.r.t. the “true” default angles  $\{\phi_i\}_i$  and the shape of the “true” resource  $|G\rangle$

# Universal Blind Quantum Computation (UBQC)

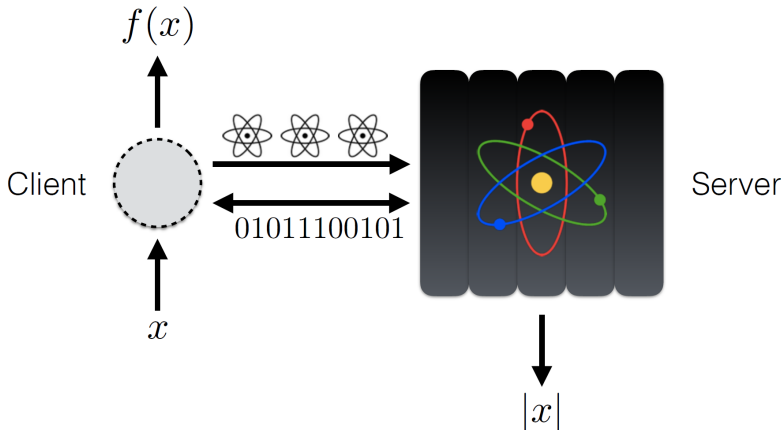
Keep  $x$  and  $f(x)$  hidden from server



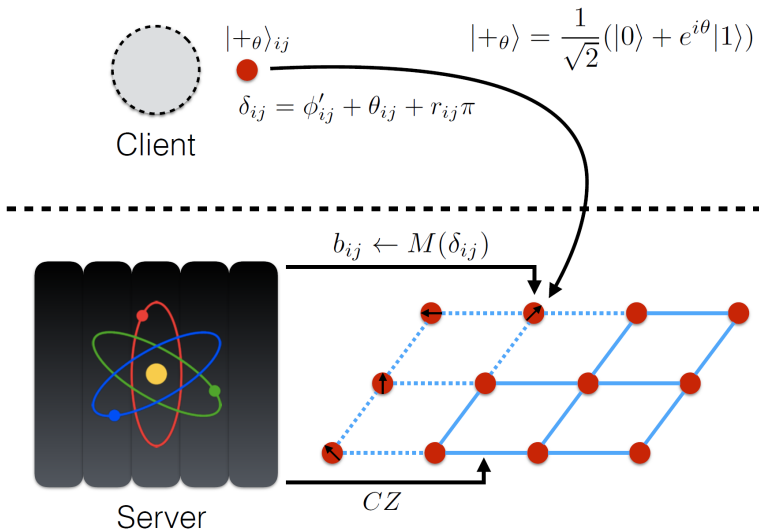
Client  $\in$  BPP

Server  $\in$  BQP

# Universal Blind Quantum Computation (UBQC)



# Universal Blind Quantum Computation (UBQC)



Broadbent, Fitzsimons, Kashefi - FOCS 2009

# Trick 1: Hiding the measurement angle

- **Properties:**

(a)  $R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2) = R(\theta_2)R(\theta_1)$ .

Rotations (on same axis), commute and act additively

# Trick 1: Hiding the measurement angle

- **Properties:**

(a)  $R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2) = R(\theta_2)R(\theta_1)$ .

Rotations (on same axis), commute and act additively

(b)  $|+\theta\rangle = R(\theta)|+\rangle = R(\theta)H|0\rangle$ .

Preparing “ $+\theta$ ” states from “0”.

$$|-\theta\rangle = R(\theta)|-\rangle = R(\theta)H|1\rangle.$$

# Trick 1: Hiding the measurement angle

- **Properties:**

(a)  $R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2) = R(\theta_2)R(\theta_1)$ .

Rotations (on same axis), commute and act additively

(b)  $|+\theta\rangle = R(\theta)|+\rangle = R(\theta)H|0\rangle$ .

Preparing “ $+\theta$ ” states from “0”.

$$|-\theta\rangle = R(\theta)|-\rangle = R(\theta)H|1\rangle.$$

(c)  $M^\alpha = M^ZHR(-\alpha)$ .

Measuring at an angle is equivalent with applying the inverse circuit that prepares  $|\pm_\alpha\rangle$  and then measure in comp. basis.



# Trick 1: Hiding the measurement angle

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2 = |s_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

# Trick 1: Hiding the measurement angle

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2 = |s_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

- **Scenario 2** (Input that is  $\theta$  pre-rotated)

$$M_1^{(\phi+\theta)} \wedge Z_{12} |+\theta\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi - \theta) |+\theta\rangle_2$$

$$|s_1\rangle_1 X^{s_1} HR(-\phi - \theta) R(\theta) |+\rangle = |s_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

# Trick 1: Hiding the measurement angle

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2 = |s_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

- **Scenario 2** (Input that is  $\theta$  pre-rotated)

$$M_1^{(\phi+\theta)} \wedge Z_{12} |+\theta\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi - \theta) |+\theta\rangle_2$$

$$|s_1\rangle_1 X^{s_1} HR(-\phi - \theta) R(\theta) |+\rangle = |s_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

- Two scenarios have **same** effect
- If  $\theta$  is unknown to Bob, when he measures  $(\phi + \theta)$  he is ignorant of the “true” angle of the  $J$ -gate he implements.

# Trick 2: Hiding the measurement outcome

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2$$

# Trick 2: Hiding the measurement outcome

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2$$

- **Scenario 2** (outcome hidden by  $r \leftarrow \{0, 1\}$ )

$$M_1^{\phi+r\pi} \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |b_1\rangle_1 X^{b_1} J(-\phi - r\pi) |+\rangle_2$$

$$|b_1\rangle_1 X^{b_1} HR(-r\pi)R(-\phi) |+\rangle_2 = |b_1\rangle_1 X^{b_1} HZ^r R(-\phi) |+\rangle_2$$

$$|b_1\rangle_1 X^{b_1+r} HR(-\phi) |+\rangle_2 = |b_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

where  $s_1 := b_1 + r$

# Trick 2: Hiding the measurement outcome

Consider Two Scenarios:

- **Scenario 1** (normal MBQC)

$$M_1^\phi \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |s_1\rangle_1 X^{s_1} J(-\phi) |+\rangle_2$$

- **Scenario 2** (outcome hidden by  $r \leftarrow \{0, 1\}$ )

$$M_1^{\phi+r\pi} \wedge Z_{12} |+\rangle_1 |+\rangle_2 \rightarrow |b_1\rangle_1 X^{b_1} J(-\phi - r\pi) |+\rangle_2$$

$$|b_1\rangle_1 X^{b_1} HR(-r\pi)R(-\phi) |+\rangle_2 = |b_1\rangle_1 X^{b_1} HZ^r R(-\phi) |+\rangle_2$$

$$|b_1\rangle_1 X^{b_1+r} HR(-\phi) |+\rangle_2 = |b_1\rangle_1 X^{s_1} HR(-\phi) |+\rangle_2$$

where  $s_1 := b_1 + r$

- Two scenarios have **same** effect on qubit 2
- If Bob doesn't know  $r$ , when he measures  $\phi + r\pi$  he is ignorant of the "true" measurement outcome  $s_1$  and how to correct in the future angles.

# Hiding measurement angle & outcome

- Alice sends  $|+\theta\rangle$  instead of  $|+\rangle$  to Bob ( $\theta$  unknown to Bob)

# Hiding measurement angle & outcome

- Alice sends  $|+\theta\rangle$  instead of  $|+\rangle$  to Bob ( $\theta$  unknown to Bob)
- Let  $\phi_i$  be the default and  $\phi'_i = (-1)^{s_x} \phi_i + \pi(\sum_{j \in S_Z} s_j)$  be the corrected measurement angles
- Define angle:

$$\delta_i = \phi'_i + \theta_i + r_i \pi$$

where  $r_i \leftarrow \{0, 1\}$  is unknown to Bob



# Hiding measurement angle & outcome

- Alice sends  $|+\theta\rangle$  instead of  $|+\rangle$  to Bob ( $\theta$  unknown to Bob)
- Let  $\phi_i$  be the default and  $\phi'_i = (-1)^{s_i} \phi_i + \pi(\sum_{j \in S_Z} s_j)$  be the corrected measurement angles
- Define angle:

$$\delta_i = \phi'_i + \theta_i + r_i \pi$$

where  $r_i \leftarrow \{0, 1\}$  is unknown to Bob

- Measuring  $\phi'_i$  angle on state  $|+\rangle$  is the same as measuring  $\phi'_i + \theta_i$  angle on state  $|+\theta_i\rangle$
- Adding  $r_i \pi$  does **not** change the measurement, only flips the outcome ( $s_i = 0$  goes to  $s_i = 1$  and visa-versa)

## Summary of Instructions

- Alice sends  $|+\theta_i\rangle$  and instructs Bob to measure in  $\delta_i$
- Bob returns outcome  $b_i$
- Alice computes  $s_i = b_i \oplus r_i$  and uses this for  $\phi'_j | j \in \{ \text{future of } i \}$

# Hiding measurement angle & outcome

## Summary of Instructions

- Alice sends  $|+\theta_i\rangle$  and instructs Bob to measure in  $\delta_i$
- Bob returns outcome  $b_i$
- Alice computes  $s_i = b_i \oplus r_i$  and uses this for  $\phi'_j | j \in \{ \text{future of } i \}$

The pre-rotation  $\theta_i$  one-time-pads the true measurement angle  $\phi'_i$ , and  $r_i$  one-time-pads the true measurement outcome  $s_i$

Bob is **blind** to both measurement angle (computation) *and* measurement outcome!

## Summary of Instructions

- Alice sends  $|+\theta_i\rangle$  and instructs Bob to measure in  $\delta_i$
- Bob returns outcome  $b_i$
- Alice computes  $s_i = b_i \oplus r_i$  and uses this for  $\phi'_j | j \in \{\text{future of } i\}$

The pre-rotation  $\theta_i$  one-time-pads the true measurement angle  $\phi'_i$ , and  $r_i$  one-time-pads the true measurement outcome  $s_i$

Bob is **blind** to both measurement angle (computation) *and* measurement outcome!

**Note:** Interaction is required so that Alice can compute the corrected measurement angle  $\phi'_i$  which depends on the (corrected) measurement outcomes (and thus cannot be computed by Bob).

## Trick 3: Hiding the shape of the graph state

- The shape of the resource state used may leak information about the specific computation

# Trick 3: Hiding the shape of the graph state

- The shape of the resource state used may leak information about the specific computation

## **Solution:**

- A general graph (e.g. 2-dim lattice) where the actual graph used can be embedded
- A trick to “break” the graph at a vertex (remove vertex and break connectivity of the graph)

## Trick 3: Hiding the shape of the graph state

- The shape of the resource state used may leak information about the specific computation

### **Solution:**

- A general graph (e.g. 2-dim lattice) where the actual graph used can be embedded
- A trick to “break” the graph at a vertex (remove vertex and break connectivity of the graph)
- Alternatively, could consider certain graph states that are universal with only  $|\pm\phi\rangle$  measurements (e.g. “brickwork state”)

## Trick 3: Hiding the shape of the graph state

- $\wedge Z_{12} |0\rangle_1 |\psi\rangle_2 = |0\rangle_1 |\psi\rangle_2$  ;  $\wedge Z_{12} |1\rangle_1 |\psi\rangle_2 = |1\rangle_1 (Z |\psi\rangle_2)$

Computational basis qubits do **not** get entangled with  $\wedge Z$

$$\wedge Z_{12} |d\rangle_1 |\psi\rangle_2 = |d\rangle_1 (Z^d |\psi\rangle_2)$$



# Trick 3: Hiding the shape of the graph state

- $\wedge Z_{12} |0\rangle_1 |\psi\rangle_2 = |0\rangle_1 |\psi\rangle_2$  ;  $\wedge Z_{12} |1\rangle_1 |\psi\rangle_2 = |1\rangle_1 (Z |\psi\rangle_2)$

Computational basis qubits do **not** get entangled with  $\wedge Z$

$$\wedge Z_{12} |d\rangle_1 |\psi\rangle_2 = |d\rangle_1 (Z^d |\psi\rangle_2)$$

- **Scenario 1:** Alice sends randomly  $\{|+\rangle, |-\rangle\}$  state, with equal probability

$$\text{Bob's view: } \rho = \frac{1}{2} (|+\rangle \langle +| + |-\rangle \langle -|) = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- **Scenario 2:** Alice sends randomly  $\{|0\rangle, |1\rangle\}$  state, with equal probability

$$\text{Bob's view: } \rho = \frac{1}{2} (|0\rangle \langle 0| + |1\rangle \langle 1|) = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

## Trick 3: Hiding the shape of the graph state

- Alice can send random computational basis qubits  $|d\rangle$ , instead of  $|+\theta\rangle$  for vertices that she wants the graph to “break”

These are called “**dummy**” qubits

## Trick 3: Hiding the shape of the graph state

- Alice can send random computational basis qubits  $|d\rangle$ , instead of  $|+\theta\rangle$  for vertices that she wants the graph to “break”

These are called “**dummy**” qubits

- Bob cannot distinguish the positions that the graph breaks from other positions, thus he is **ignorant of the “true” shape of the resource used**

## Trick 3: Hiding the shape of the graph state

- Alice can send random computational basis qubits  $|d\rangle$ , instead of  $|+\theta\rangle$  for vertices that she wants the graph to “break”

These are called “**dummy**” qubits

- Bob cannot distinguish the positions that the graph breaks from other positions, thus he is **ignorant of the “true” shape of the resource used**
- The dummy qubits produce a  $Z^d$  **correction** to all neighbouring qubits.

These corrections are:

(i) known to Alice only, (ii) know from the start

Alice takes them into account when computing the angle that she asks Bob to measure

(Adds  $d\pi$  to the angle of qubits neighbouring with  $|d\rangle$  qubit)

## UBQC protocol and Verification of Quantum Computing

# A first UBQC protocol

We assume only  $M^\alpha$  measurements (breaking to the desired resource state happens as described)

We assume classical input/output (can generalise)

## Input:

- Graph  $G$  of  $m$  qubits sufficient for given computation
- $m$  “default” measurement angles  $\phi_i$  performing desired computation
- $m$  random variables  $\theta_i \leftarrow \{0, \pi/4, \dots, 7\pi/4\}$ ,  $m$  random variables  $r_i \leftarrow \{0, 1\}$  chosen secretly by Alice

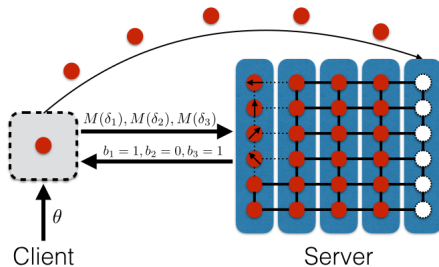
## Initial Step:

- Alice sends to Bob  $m$  qubits of the form  $|+\theta_i\rangle$
- Bob applies  $\wedge Z_{ij}$  according to the graph and generates the “secretly rotated” resource state

# A first UBQC protocol

## Step $i$ : $1 \leq i \leq m$

- Alice computes the angle  $\delta_i = \phi'_i + \theta_i + r_i\pi$  and instructs Bob to measure qubit  $i$  at this angle
- Bob measures qubit  $i$  and returns outcome  $b_i$  to Alice
- Alice sets the value  $s_i = b_i \oplus r_i$
- Alice moves to step  $i + 1$  until  $i = m$  where the protocol terminates
- The outcome is obtained from the last “layer” of measurements



# Towards Verification of Quantum Computation

- Consider qubit  $i$  where **all** neighbours are dummies.
- Qubit  $i$  is disentangled from the rest graph



# Towards Verification of Quantum Computation

- Consider qubit  $i$  where **all** neighbours are dummies.
- Qubit  $i$  is disentangled from the rest graph
- Is called **trap qubit** and is at the state:

$$Z^{\sum_{j \in N_G(i)} d_j} |+\theta_i\rangle_i$$

# Towards Verification of Quantum Computation

- Consider qubit  $i$  where **all** neighbours are dummies.
- Qubit  $i$  is disentangled from the rest graph
- Is called **trap qubit** and is at the state:

$$Z^{\sum_{j \in N_G(i)} d_j} |+\theta_i\rangle_i$$

- If measured in the  $\{|+\theta_i\rangle, |-\theta_i\rangle\}$  basis it will (deterministically) give the outcome:

$$b_i := \sum_{j \in N_G(i)} d_j$$

# Towards Verification of Quantum Computation

- Consider qubit  $i$  where **all** neighbours are dummies.
- Qubit  $i$  is disentangled from the rest graph
- Is called **trap qubit** and is at the state:

$$Z^{\sum_{j \in N_G(i)} d_j} |+\theta_i\rangle_i$$

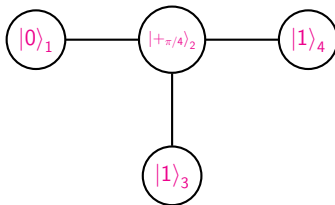
- If measured in the  $\{|+\theta_i\rangle, |-\theta_i\rangle\}$  basis it will (deterministically) give the outcome:

$$b_i := \sum_{j \in N_G(i)} d_j$$

- Alice knows this result in advance, but Bob doesn't  
(Bob neither knows the  $d$ 's nor the position that a trap exists)

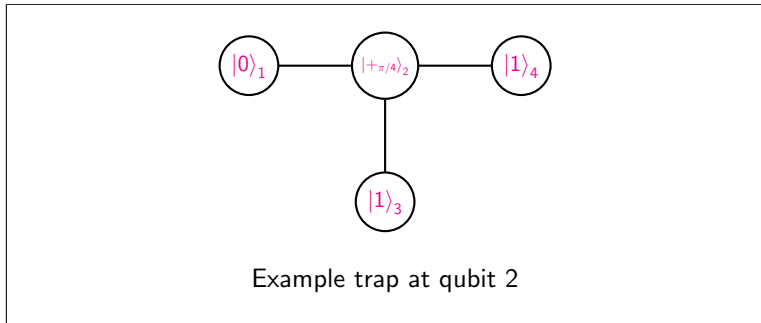
# Towards Verification of Quantum Computation

**An example:** with  $d_1 = 0, \theta_2 = \pi/4, d_3 = 1, d_4 = 1$



Example trap at qubit 2

An example: with  $d_1 = 0, \theta_2 = \pi/4, d_3 = 1, d_4 = 1$



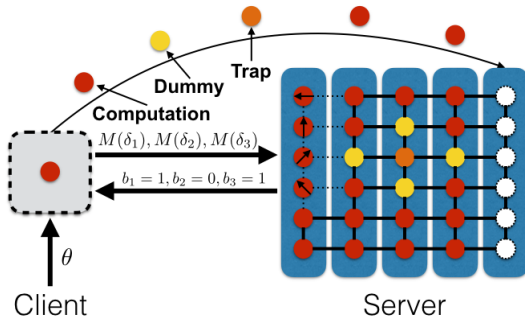
- The trap qubit (after  $\wedge Z$ -gates) is at state:

$$Z^{d_1+d_3+d_4} |+\pi/4\rangle = Z^2 |+\pi/4\rangle = |+\pi/4\rangle$$

- If measured in the  $\{|+\pi/4\rangle, |-\pi/4\rangle\}$ -basis we get (always)  
 $b_2 = 0$

# Towards Verification of Quantum Computation

- **Position** of traps and dummies is **unknown** to Bob
- **Result of measurement of trap**, if measured in  $M_t^\theta$ -basis is deterministic and **known** in advance to **Alice**
- **If Bob deviates** at the protocol, he may deviate on the trap qubit and this will be **detected by Alice!**



# Towards Verification of Quantum Computation

- Can insert **multiple independent traps** without revealing anything or disrupting the computation

# Towards Verification of Quantum Computation

- Can insert **multiple independent traps** without revealing anything or disrupting the computation
- Computation encoded with **QECC**.
- **Computation**  $\Rightarrow$  logical qubits, **traps**  $\Rightarrow$  physical qubits.



# Towards Verification of Quantum Computation

- Can insert **multiple independent traps** without revealing anything or disrupting the computation
- Computation encoded with **QECC**.
- **Computation**  $\Rightarrow$  logical qubits, **traps**  $\Rightarrow$  physical qubits.
- **Single error on trap**  $\Rightarrow$  abort, **multiple errors on computation** for corrupt the computation.
- **Protocol fails with  $\epsilon$ -probability for “corrupt AND not-abort”**

# Towards Verification of Quantum Computation

- Can insert **multiple independent traps** without revealing anything or disrupting the computation
- Computation encoded with **QECC**.
- **Computation**  $\Rightarrow$  logical qubits, **traps**  $\Rightarrow$  physical qubits.
- **Single error on trap**  $\Rightarrow$  abort, **multiple errors on computation** for corrupt the computation.
- **Protocol fails with  $\epsilon$ -probability for “corrupt AND not-abort”**
- **verification (VBQC) = blindness + traps**
- Verify correctness against malicious prover  $\Rightarrow$  correctness against errors due to noise/malfunctions (even correlated)

# Towards Verification of Quantum Computation

- Can insert **multiple independent traps** without revealing anything or disrupting the computation
- Computation encoded with **QECC**.
- **Computation**  $\Rightarrow$  logical qubits, **traps**  $\Rightarrow$  physical qubits.
- **Single error on trap**  $\Rightarrow$  abort, **multiple errors on computation** for corrupt the computation.
- **Protocol fails with  $\epsilon$ -probability for “corrupt AND not-abort”**
- **verification (VBQC) = blindness + traps**
- Verify correctness against malicious prover  $\Rightarrow$  correctness against errors due to noise/malfunctions (even correlated)

**Method to verify/test any quantum computation device**

## Blind Quantum Computation

- 1 A. Broadbent, J. Fitzsimons and E. Kashefi, *Universal Blind Quantum Computation*, in Foundations of Computer Science (FOCS) 517, (2009).
- 2 Stefanie Barz, Elham Kashefi, Anne Broadbent, Joseph F. Fitzsimons, Anton Zeilinger, Philip Walther, *Demonstration of Blind Quantum Computing*, Science 335, 303 (2012).

## Verifiable Blind Quantum Computation

- 3 J. Fitzsimons and E. Kashefi, *Unconditionally verifiable blind computation*, preprint 1203.5217 (2012).
- 4 S. Barz, J. Fitzsimons, E. Kashefi and P. Walther, *Experimental verification of quantum computations*, Nature Physics, 9 727 (2013).
- 5 E. Kashefi and P. Wallden, *Optimised resource construction for verifiable quantum computation*, J. Phys. A: Math. Theor. 50 145306 (2017).