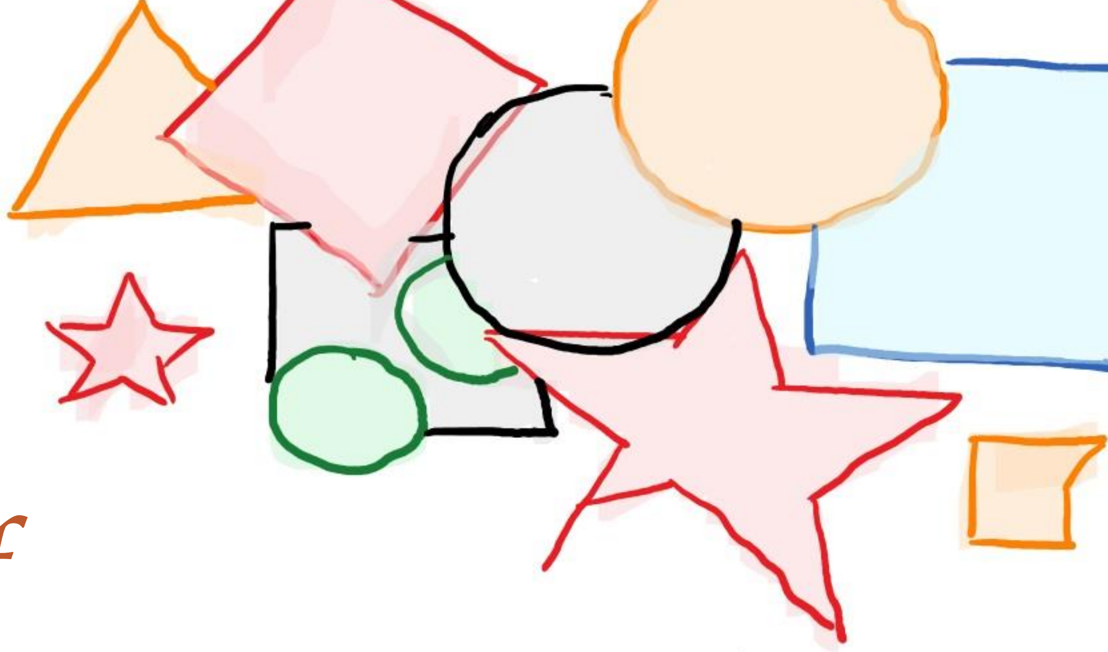
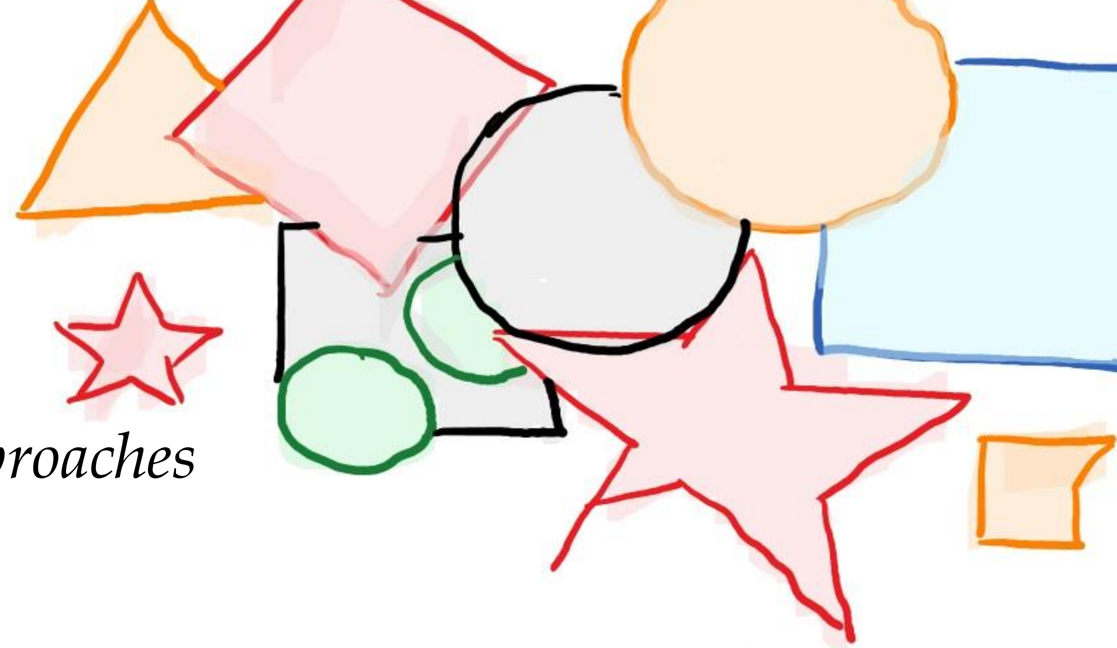


Verification of quantum computation

THOMAS VIDICK

CALIFORNIA INSTITUTE OF TECHNOLOGY

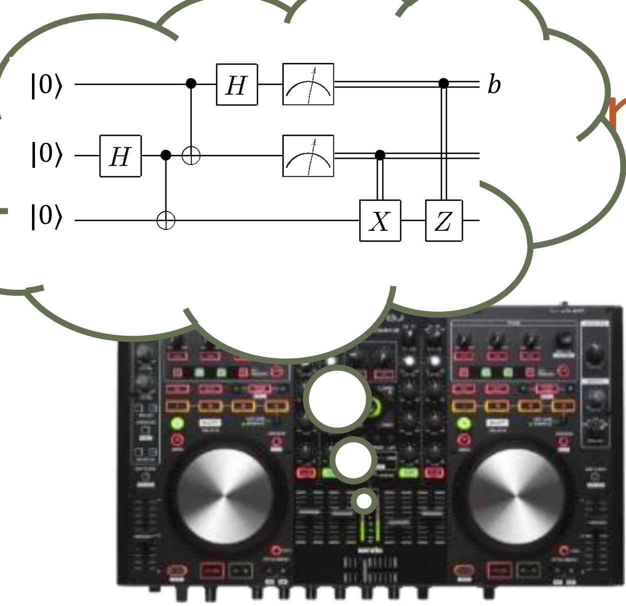
Slides: http://users.cms.caltech.edu/~vidick/verification.{ppsx,pdf}



1. *Problem formulation*
2. *Overview of existing approaches*
3. *Prepare & send*
4. *Two-prover delegation*
5. *Receive & measure*
6. *Commit & reveal*
7. *Coda*

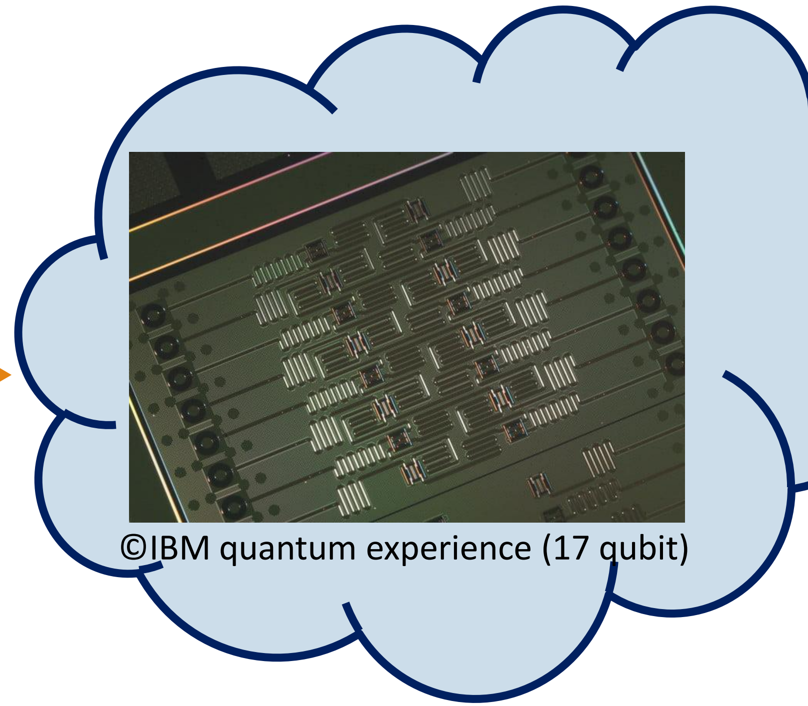
Problem formulation

mulation



classical or quantum

communication



user

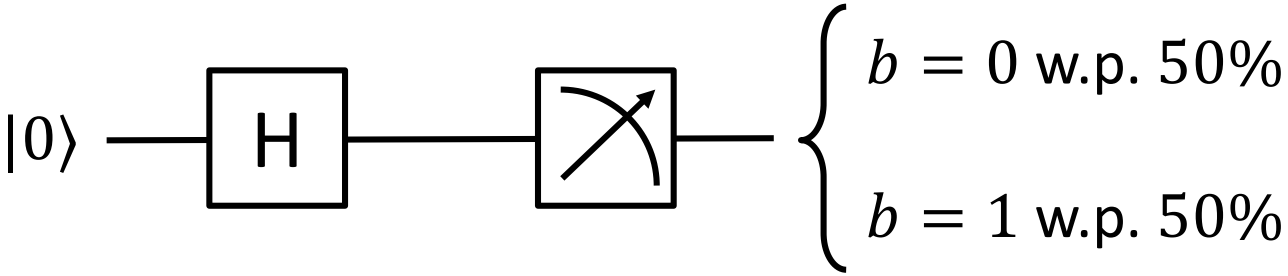
$\rightarrow (\boldsymbol{flag}, \boldsymbol{b})$

device

- Verifier has quantum computation C
- Multiple rounds of interaction with quantum device
- Verifier returns $(flag, b)$ s.t. $flag \in \{acc, rej\}$ and $b \in \{0,1\}$
- Goal: Whenever $\Pr(flag = acc)$ is non-negligible,

$$\Pr(\, b = 1 \,|flag = acc) \quad \approx \quad \Pr(\, C \text{ returns } 1 \text{ on input } |0^n\rangle \quad)$$

An example



"description of circuit C"

"I got $b = 0$"

Really??

Join the IBM Q Experience Community

The IBM Q Experience Community brings together researchers and quantum enthusiasts to share, connect and collaborate

If you want to interact within the community, you need a username.

Set your username

Post to forum

Search for...

All Categories

21 comments

4.9k views

15 likes

Software

IBM Q Awards Contest Program

Submit a contribution to the IBM Q Awards !The IBM Q Awards are a series of prizes for professors, lecturers and students who use the IBM Q Experience and QISKi...

AN **andreasf** IBM Staff Posted 10 months ago Last comment by yy387 10 days ago

IBM Q Awards | IBM QE | QISKit | quantum software | compiling

1 comments

9 views

1

Software

Results in hex format?

Does anyone else find the sudden change of presenting results in hex and not binary counterintuitive? I'm sure everyone in the field of QI is more familiar with...

XA **xavierlin** Posted a day ago Last comment by constantine 3 hours ago

Quick links

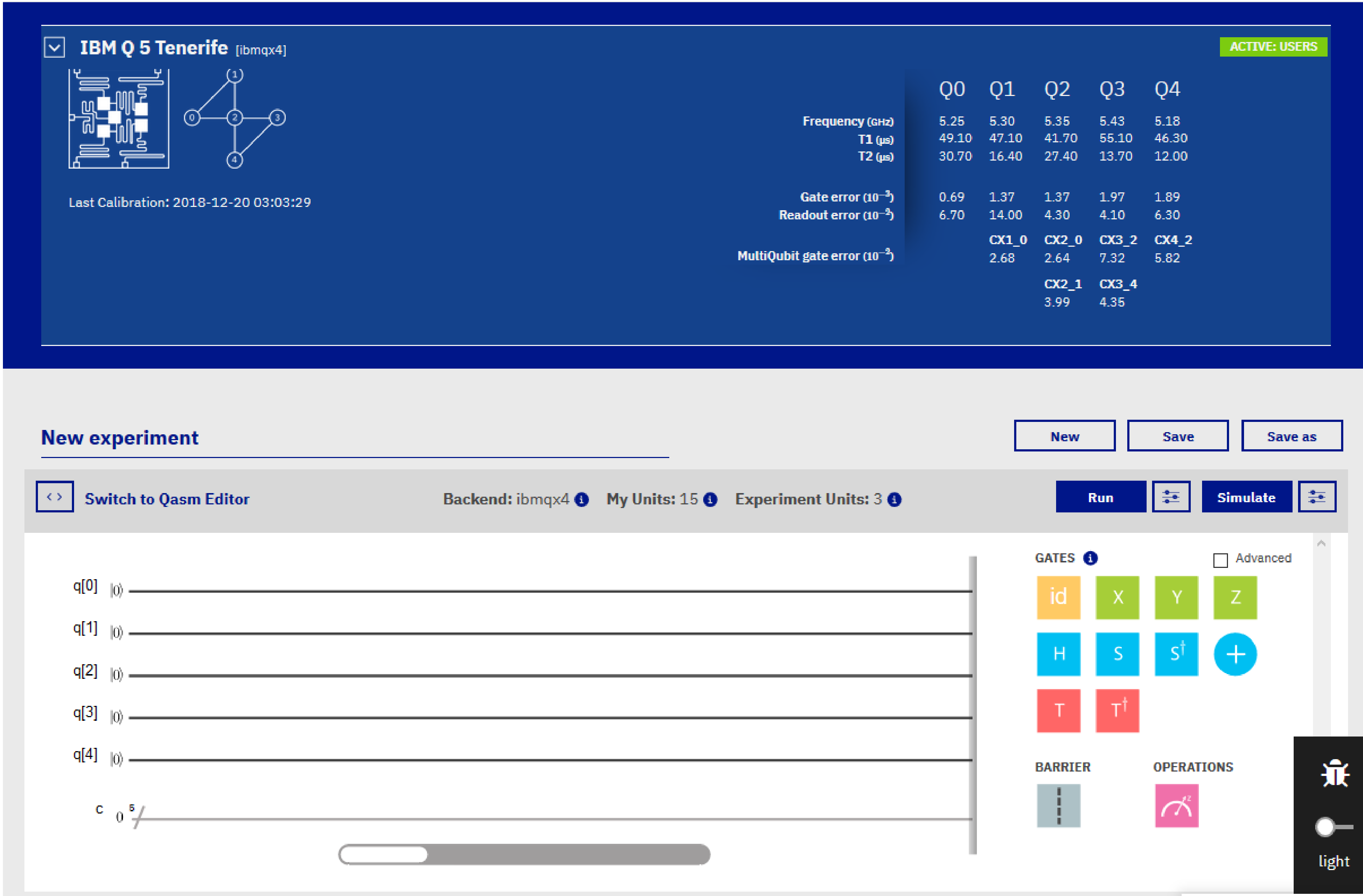
- FAQ
- Beginner's Guide
- Full User Guide

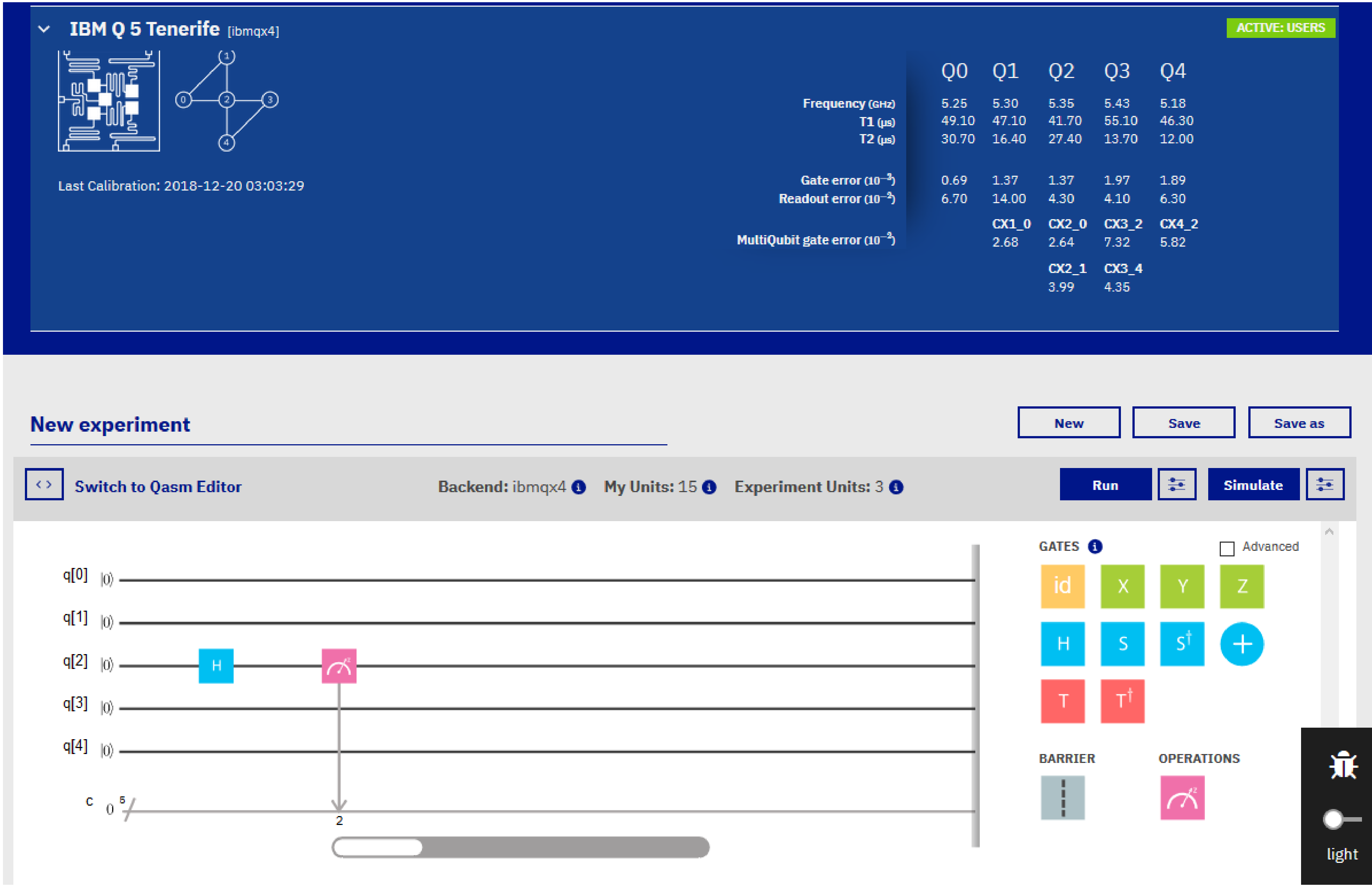
Tags

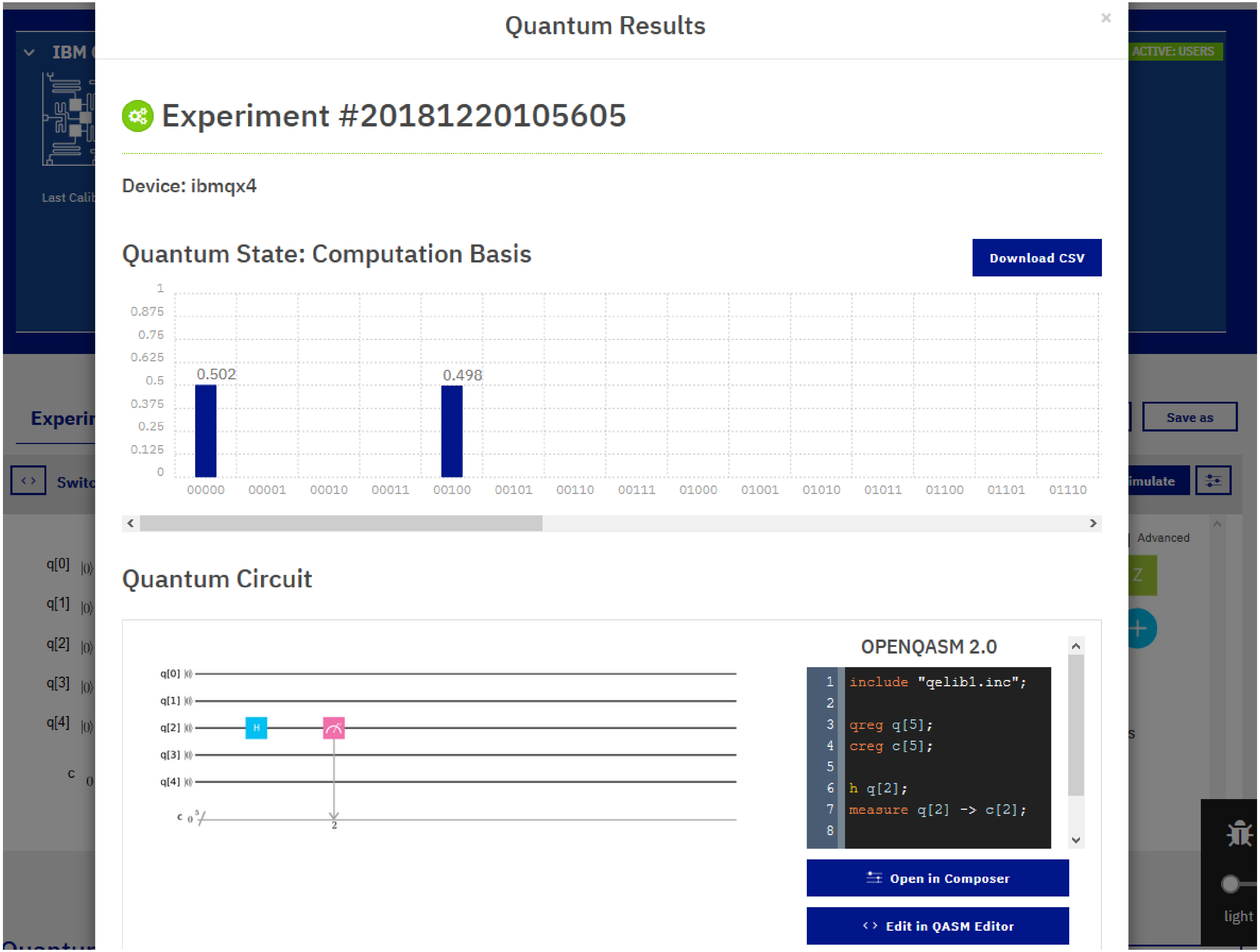
Top Users (Last week)

MR PI BI AV JU OG EV SI A_ JA LE CH

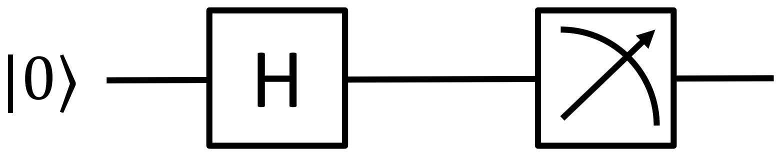
Courses


IBM Q 5 Tenerife [ibmqx4]
ACTIVE: USERS
Last Calibration: 2018-12-20 03:03:29
Q0 Q1 Q2 Q3 Q4
Frequency (GHz) 5.25 5.30 5.35 5.43 5.18
T1 (µs) 49.10 47.10 41.70 55.10 46.30
T2 (µs) 30.70 16.40 27.40 13.70 12.00
Gate error (10⁻³) 0.69 1.37 1.37 1.97 1.89
Readout error (10⁻²) 6.70 14.00 4.30 4.10 6.30
MultiQubit gate error (10⁻²)
CX1_0 2.68
CX2_0 2.64
CX3_2 7.32
CX4_2 5.82
CX2_1 3.99
CX3_4 4.35
New experiment
New
Save
Save as
Switch to Qasm Editor
Backend: ibmqx4
My Units: 15
Experiment Units: 3
Run
Simulate
q[0] |0⟩
q[1] |0⟩
q[2] |0⟩
q[3] |0⟩
q[4] |0⟩
c 0 5
GATES
Advanced
id X Y Z
H S S† +
T T†
BARRIER
OPERATIONS
light

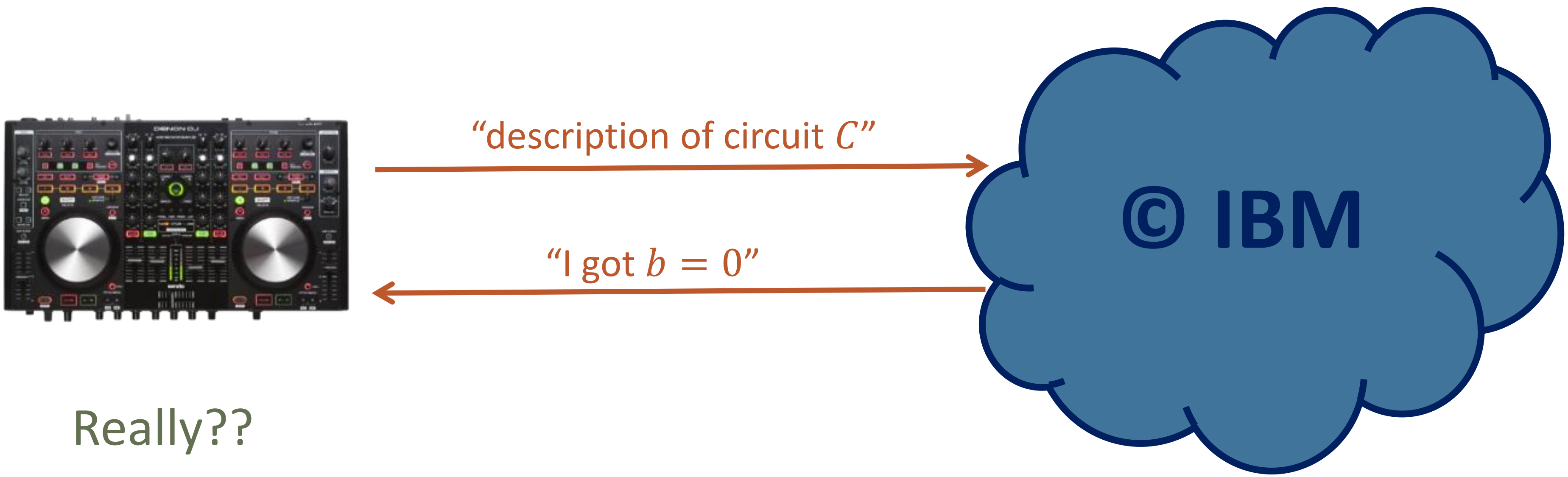

IBM Q 5 Tenerife [ibmqx4]
ACTIVE: USERS
Last Calibration: 2018-12-20 03:03:29
Q0 Q1 Q2 Q3 Q4
Frequency (GHz) 5.25 5.30 5.35 5.43 5.18
T1 (µs) 49.10 47.10 41.70 55.10 46.30
T2 (µs) 30.70 16.40 27.40 13.70 12.00
Gate error (10⁻³) 0.69 1.37 1.37 1.97 1.89
Readout error (10⁻³) 6.70 14.00 4.30 4.10 6.30
MultiQubit gate error (10⁻³)
CX1_0 2.68
CX2_0 2.64
CX3_2 7.32
CX4_2 5.82
CX2_1 3.99
CX3_4 4.35
New experiment
New
Save
Save as
Switch to Qasm Editor
Backend: ibmqx4
My Units: 15
Experiment Units: 3
Run
Simulate
q[0]
q[1]
q[2]
q[3]
q[4]
c
H
GATES
Advanced
id
X
Y
Z
H
S
S†
T
T†
BARRIER
OPERATIONS
light


Quantum Results
Experiment #20181220105605
Device: ibmqx4
Quantum State: Computation Basis
Download CSV
1
0.875
0.75
0.625
0.5
0.375
0.25
0.125
0
0.502
0.498
00000
00001
00010
00011
00100
00101
00110
00111
01000
01001
01010
01011
01100
01101
01110
Quantum Circuit
q[0]
q[1]
q[2]
q[3]
q[4]
c
OPENQASM 2.0
include "qelib1.inc";
qreg q[5];
creg c[5];
h q[2];
measure q[2] -> c[2];
Open in Composer
Edit in QASM Editor
Save as
Advanced
light

An example



$$\begin{cases} b = 0 \text{ w.p. } 50\% \\ b = 1 \text{ w.p. } 50\% \end{cases}$$

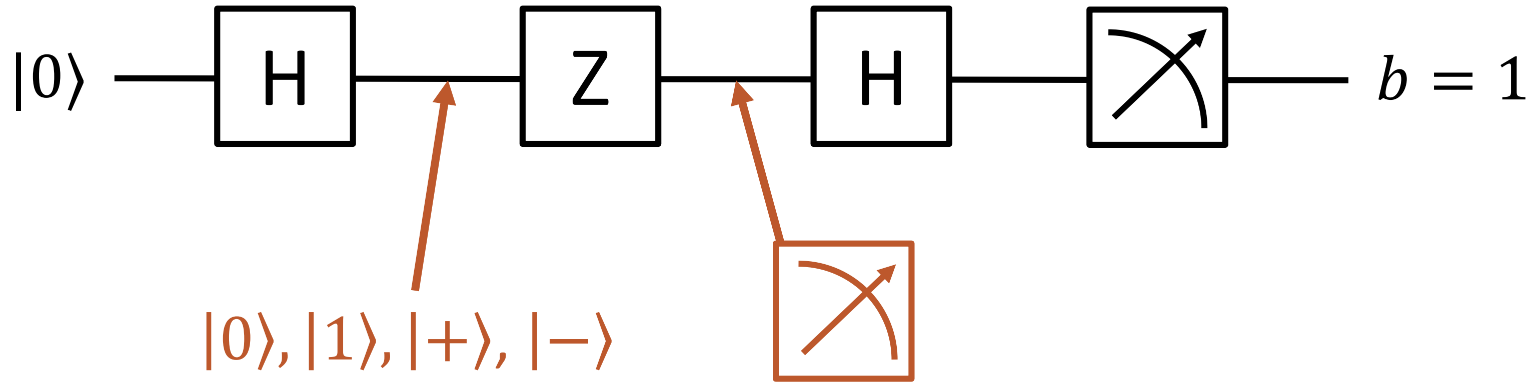


Really??

Repeat and collect statistics?

Run some tests?

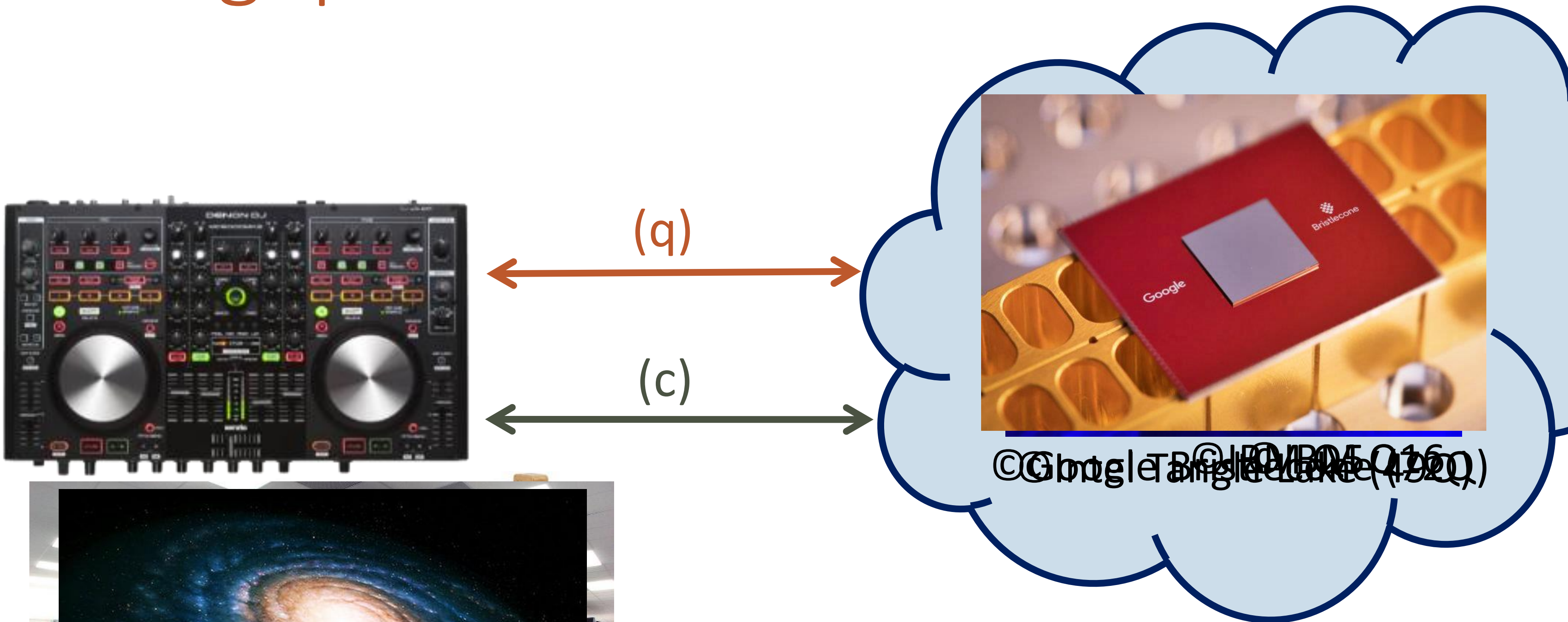
Aside: benchmarking



Sequentially test gate by injecting well-characterized states and collecting output statistics

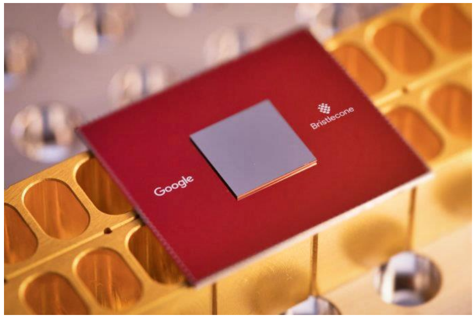
- Requires access to inner workings of device
- Trusted state preparation and/or measurement
- Gates are not allowed to be "malicious", e.g. i.i.d. behavior is generally assumed
- Ineffective at large scales

Testing quantum mechanics at scale



- Quantum mechanics untested at large scales
- Is there a limit to the exponential scaling of quantum devices?

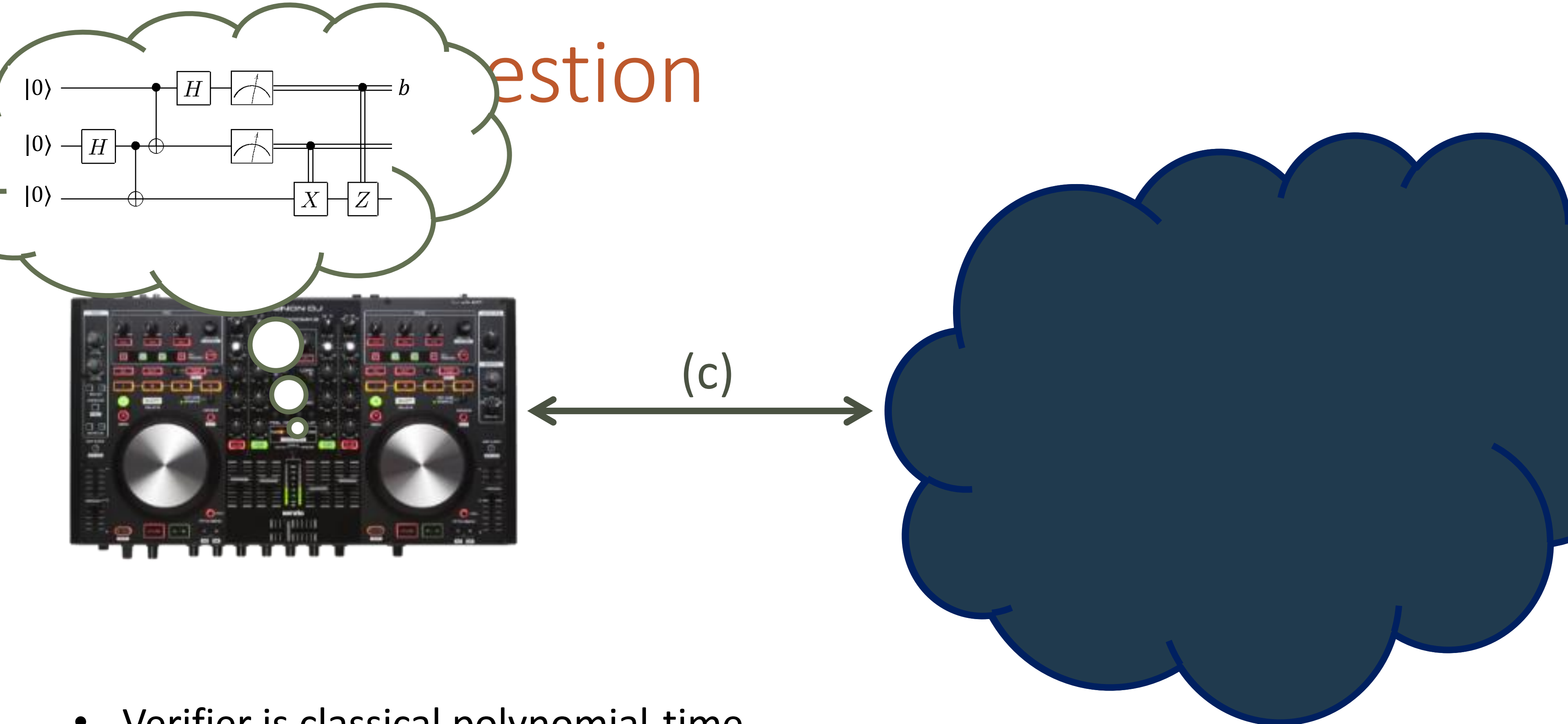
Some other reasons to care



©Google Bristlecone

- Near-term demonstration of quantum advantage
  - Can verifiability be baked in current proposals?
- Cryptographic techniques
  - What modes of encryption allow transversal computation?
  - Can they be combined with authentication?
- Models of computation & fault-tolerance
  - Do small nodes in a quantum network create fault-tolerance bottlenecks?
- Complexity theory
  - What is the expressive power of bounded-prover interactive proofs?
- Foundations
  - Are there analogues of the Bell inequalities without locality assumptions?

estion



- Verifier is classical polynomial-time
- Communication channel is classical
- Can it verify a quantum computation?

Prelude: Definitions

Informal definitions

<u>A delegation protocol for quantum computations is:</u>

A description of a (classical or quantum) polynomial-time **verifier,** that takes as input a **quantum circuit C** of size $|C| \leq n$, interacts with a **quantum prover**, and returns a pair $(flag, b)$ such that:

- *(Correctness)* There exists a (quantum, poly-time) prover P such that

  $V_n(C) \leftrightarrow P$ returns $(flag = acc, b \approx C|0\rangle)$

- *(Verifiability)* For any prover P^* such that $\Pr(flag = acc)$ is non-negligible,

  $\Pr(\, b = 1 \,|flag = \text{acc}) \quad \approx \quad \Pr(\, C \; returns \; 1 \; on \; input \; |0^n\rangle \quad)$

- *(Blindness)* For any prover P^*, $View_P(V_n(C) \leftrightarrow P^*)$ does not depend on C

Formal definitions

"Stand-alone" definitions can fail! Example:

Protocol for testing if formula $\varphi = (x_1 \vee \overline{x_3} \vee x_5) \wedge (\cdots)$ *is satisfiable*

1. Prover sends assignment $x = (x_1, \ldots, x_n)$
2. Verifier checks that x satisfies φ

This protocol is blind (prover learns nothing about φ) & verifiable

"Attack": Prover sends a uniformly random assignment

- Learns information about φ from verifier's accept/reject decision
- Protocol is not composable

Composable security: ideal-world/real-world paradigm

Formal definitions

Ideal functionality for verifiable & blind delegation

circuit C

bit b

$$b = \begin{cases} C(|0\rangle) & \text{if } e \in \{0,1\} \\ err & \text{if } e = 2 \end{cases}$$

size T

trit e

$e = 0$: honest behavior

$e \in \{1,2\}$: dishonest behavior

Composable definition (informal):

A protocol is verifiable & blind if for each party there exists a *simulator* such that for any malicious party the interaction (honest party)↔(malicious party) is indistinguishable from the interaction (ideal functionality) ↔(simulator) ↔(malicious party)

[DFPR'13] Many, *but not all*, of the protocols presented today are composable

Parameters

Input size: n = number of qubits of circuit C

$|C|$ = number of gates

Completenes: Probability of accepting honest prover. This will always be ≈ 1

Soundness: Max. distinguishing ability between real-world/ideal-world.

Ideally, exponentially small in n.

Verifier complexity: Ideally, classical polynomial-time.

Limited quantum capability may be acceptable.

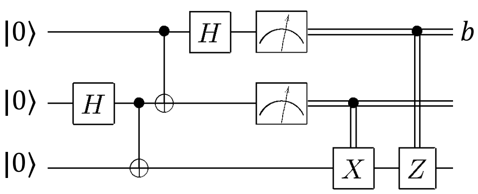
Prover complexity: Quantum polynomial-time. Ideally $\approx$ runtime(C).

Interaction: Minimize number of rounds + total communication

Overview of existing approaches

Models of computation

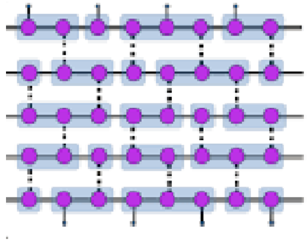
Circuit model



Input: circuit = sequence of gates acting on n qubits

Goal: determine value of output qubit, on input $|0\rangle$

Measurement-based



Input: adaptive sequence of single-qubit measurements

on resource state (e.g. "cluster state")

Goal: determine value of output qubit

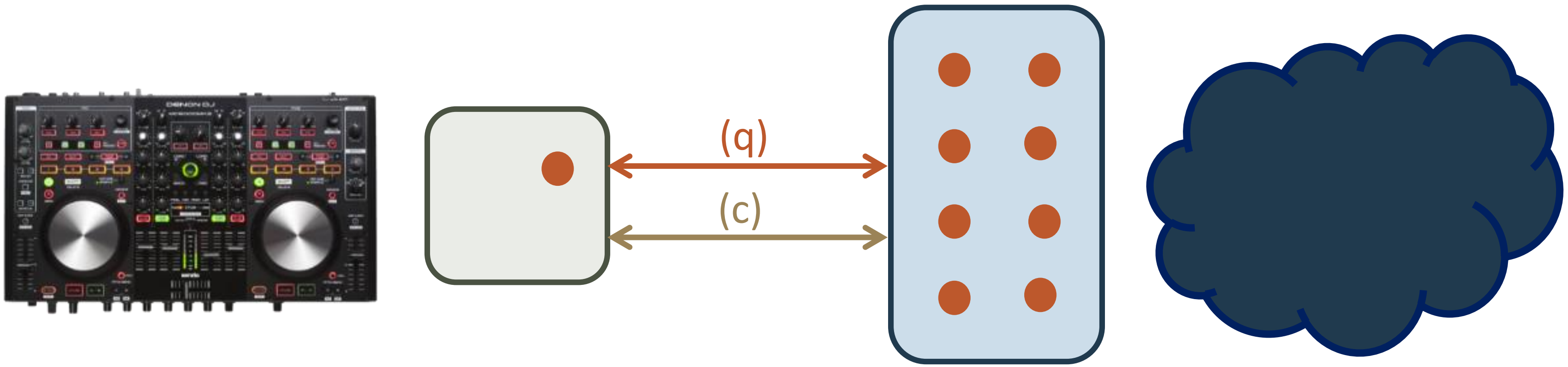
Hamiltonian model

$$H = H_{in} + H_{clock} + H_{prop} + H_{out}$$

Input: local Hamiltonian w. efficiently preparable ground state

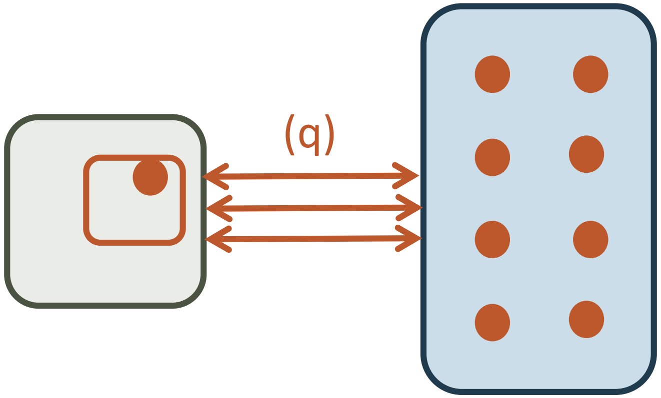
Goal: estimate ground state energy

Models for black-box verification



Challenge: Use minimal resources to verify complex quantum computation

Models for black-box verification



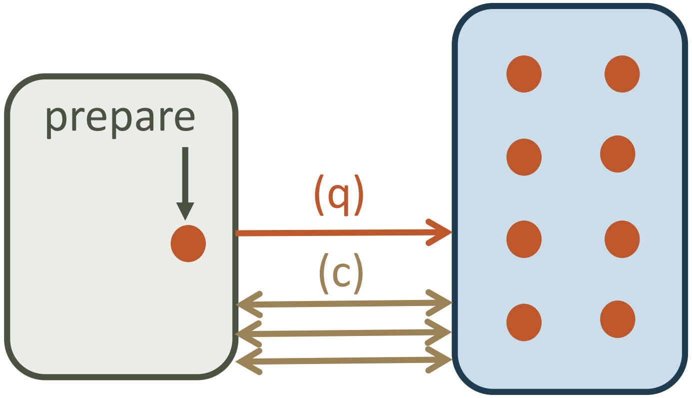
[Childs'05] Blind delegation

- Verifier has constant-size quantum computer and can only perform single-qubit Pauli gates
- Many-round quantum interaction
- Blind but not verifiable

Where are the qubits?

Honest-but-curious model

Models for black-box verification



[Aharonov-Ben-Or-Eban'08, Aharonov-Ben-Or-Eban-Mahadev'18]

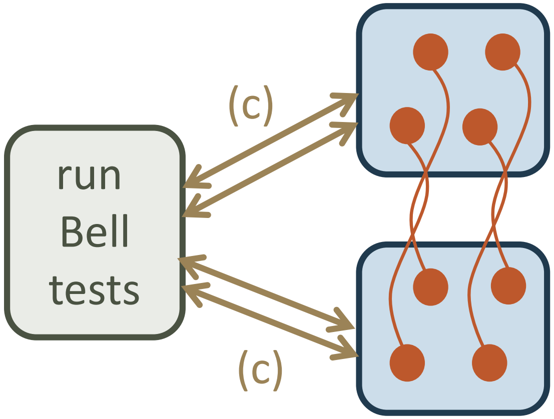
[Broadbent-Fitzsimons-Kashefi'09,Fitzsimons-Kashefi'16]

"Prepare-and-send" protocols:

- Verifier has ability to prepare & send O(1) qubits at a time
- Many-round classical interaction
  - [ABOE] *Circuit model*, uses authentication codes
  - [BFK] *Measurement-based model*, uses traps
- Both protocols are blind + verifiable

Where are the qubits? *The verifier creates them*

Models for black-box verification



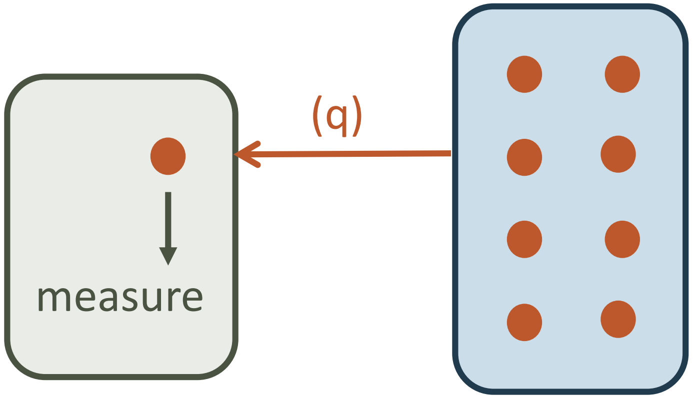
[Reichardt-Unger-Vazirani'12]

Two-prover protocols:

- Verifier is classical
- Many-round classical interaction with two isolated provers
- Verifier uses Bell tests to do state & process tomography
- Protocol is blind + verifiable

Where are the qubits? *Bell tests* $\rightarrow$ *EPR pairs* $\rightarrow$ *qubits*

Models for black-box verification



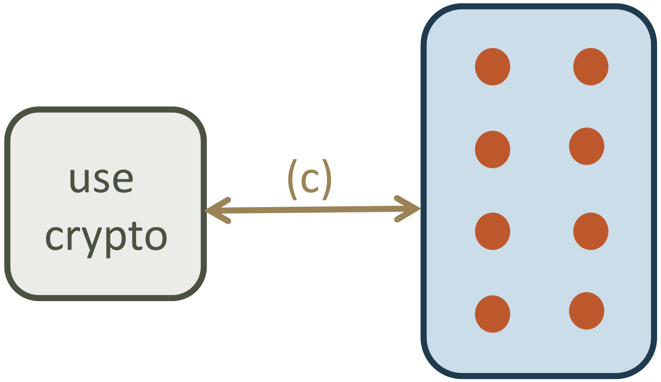
[Morimae-Fuji’13, Morimae-Fitzsimons’16]

“Receive & measure” protocols:

- Verifier has ability to receive & measure constant qubits
- [MF’13] *Measurement-based model*, protocol is blind & verifiable
- [MF’16] *Hamiltonian model*, protocol is verifiable but not blind

Where are the qubits? *The verifier measures them*

Models for black-box verification

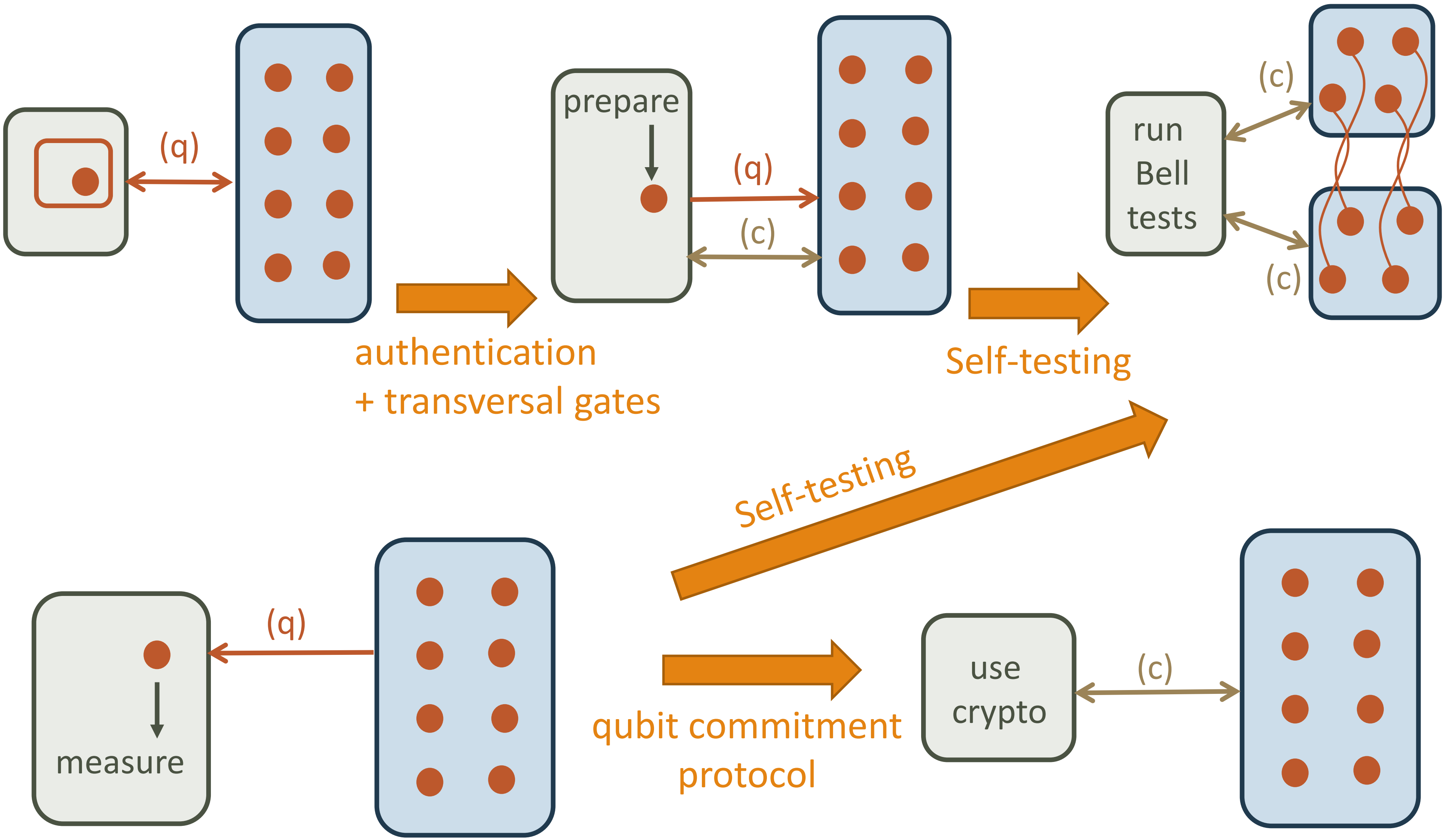


[Mahadev'18] "Commit & Reveal" protocols:

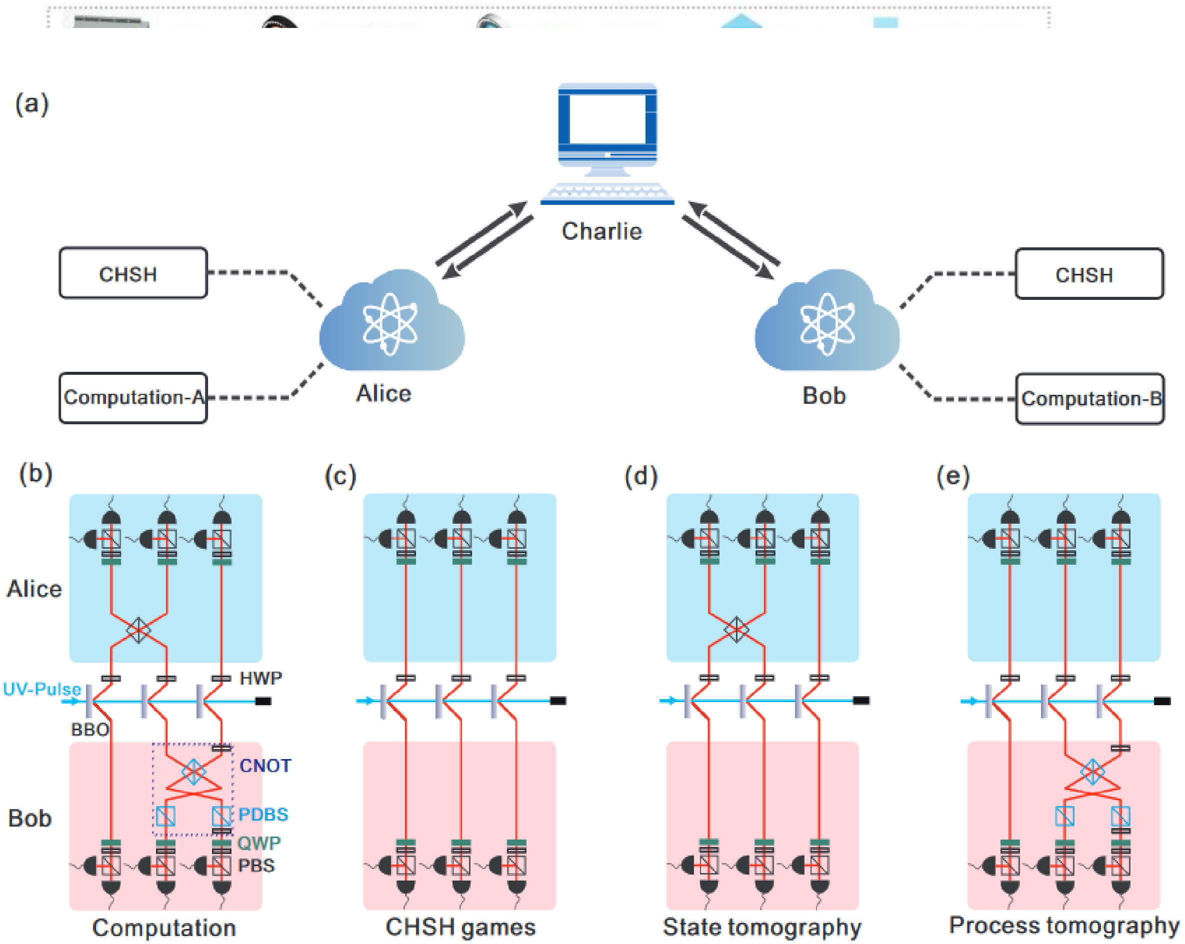
- Verifier is classical
- *Hamiltonian model*: protocol is not blind
- Verifiability assumes prover does not break post-quantum crypto

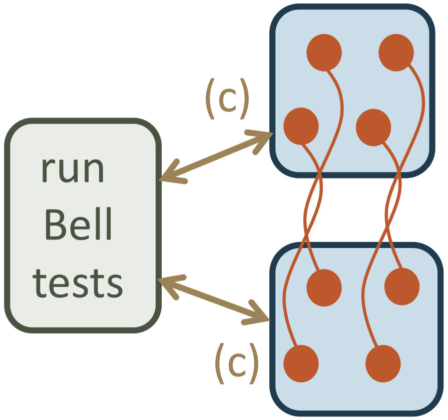
Where are the qubits? *Forced by the crypto*

ROADMAP


(q)
prepare
(q)
(c)
run
Bell
tests
(c)
(c)
authentication
+ transversal gates
Self-testing
Self-testing
(q)
measure
qubit commitment
protocol
use
crypto
(c)

Some experiments

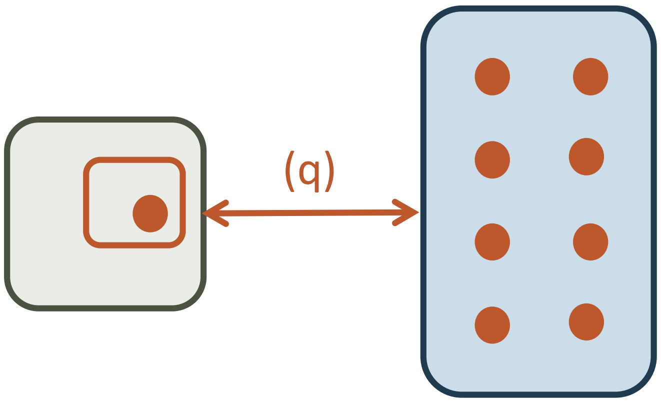




[Huang et al. 2017] Thousands of Bell tests certify factorization of number 15

Part I(a): Prepare & Send

Blind delegated computation
[Childs'05]



- Circuit model: verifier has circuit C, wants to determine outcome on $|0\rangle$
- Encode computation in input: execute universal circuit $\mathcal{U}$ to obtain $C|0\rangle$
- Main technique is "computation on encrypted data":
  - Verifier encrypts input qubits one-by-one and sends to prover
  - Prover stores qubits & applies gates over encryption
  - For each gate, verifier requests qubits, "fixes encryption", and re-sends

The quantum one-time pad

$$|\psi\rangle \quad \mapsto \quad X^a Z^b |\psi\rangle$$

$$a, b \leftarrow_R \{0,1\}$$

$$\alpha|0\rangle + \beta|1\rangle \quad \mapsto \quad \begin{cases} \alpha|0\rangle + \beta|1\rangle & (a = 0, b = 0) \\ \alpha|0\rangle - \beta|1\rangle & (a = 0, b = 1) \\ \alpha|1\rangle + \beta|0\rangle & (a = 1, b = 0) \\ \alpha|1\rangle - \beta|0\rangle & (a = 1, b = 1) \end{cases}$$

- If a, b unknown then encoded qubit appears totally mixed

$$\frac{1}{4}\sum_{a,b}(X^a Z^b)|\psi\rangle\langle\psi|(X^a Z^b)^* = \frac{1}{2}\mathbb{I}$$

Computing on encrypted data

- Clifford gates "commute" with one-time pad

Ex: $HX^aZ^b|\psi\rangle = X^bZ^aH|\psi\rangle$

- Universal computation requires one additional gate

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \qquad P = T^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

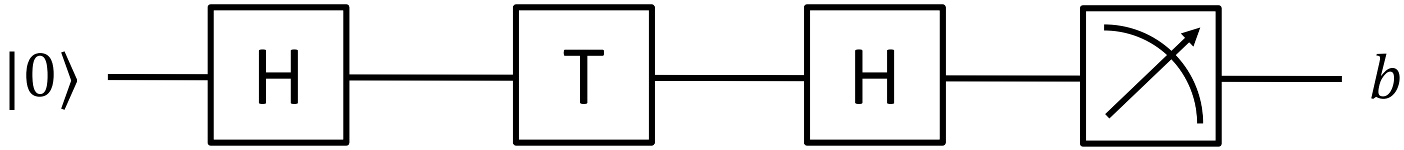
$$TX^aZ^b|\psi\rangle = X^{a'}Z^{b'}P^cT|\psi\rangle$$

- Requires "phase correction" if $c' = c'(a', b', c) = 1$

$$P^{c'}X^{a'}Z^{b'}P^cT|\psi\rangle = X^{a''}Z^{b''}T|\psi\rangle$$

- Eastin Knill theorem: no quantum error-correcting code can transversally implement a quantum universal gate set

Running example



$(a, b) \leftarrow_R \{0,1\}$

$X^a Z^b |0\rangle$

$X^a Z^b |0\rangle$ →

$X^a Z^b |0\rangle \mapsto H X^a Z^b |0\rangle$

← $H X^a Z^b |0\rangle$

$(a', b') \leftarrow_R \{0,1\}$

$H X^a Z^b |0\rangle = X^b Z^a H |0\rangle \mapsto X^{a'} Z^{b'} H |0\rangle$

$X^{a'} Z^{b'} H |0\rangle$ →

$X^{a'} Z^{b'} H |0\rangle \mapsto T X^{a'} Z^{b'} H |0\rangle$

← $T X^{a'} Z^{b'} H |0\rangle$

$T X^{a'} Z^{b'} H |0\rangle = X^{a''} Z^{b''} P^c T H |0\rangle$

$X^{a''} Z^{b''} P^c T H |0\rangle, c'$ →

$X^{a''} Z^{b''} P^c T H |0\rangle \mapsto P^{c'} X^{a''} Z^{b''} P^c T H |0\rangle$

Authentication

$2k$ "trap qubits"

$$|\psi\rangle$$

$$\mapsto \quad Q\,(|\psi\rangle|0\rangle\cdots|0\rangle|+\rangle\cdots|+\rangle)$$

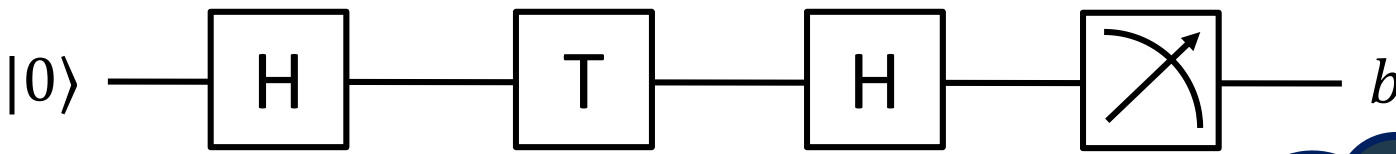
$Q \leftarrow_R \ (2k+1)$−qubit Clifford

- Random Clifford subsumes one-time pad: automatically blind
- *Clifford twirl:* any unitary "attack" independent of Q induces a random Pauli "attack" on the trap qubits

For any unitary U and any density ρ,

$$\frac{1}{|Cliff|}\sum_{C} Q^*\, U\,(Q\,\rho\,Q^*)\,U^*\,Q \;=\alpha\,\rho+\frac{1-\alpha}{|Pauli|-1}\sum_{P:pauli\neq I} P\;\rho\;P^*$$

Running example



$(a_1, b_1), (a_2, b_2),$
$(a_3, b_3) \leftarrow_R \{0,1\}$

$X^{a_1} Z^{b_1} |0\rangle X^{a_2} |0\rangle Z^{b_3} |+\rangle$ →

$X^{a_1} Z^{b_1} |0\rangle X^{a_2} |0\rangle Z^{b_3} |+\rangle$

← $X^{a_1} Z^{b_1} |0\rangle X^{a_2} |0\rangle Z^{b_3} |+\rangle$

Decode
Check traps
Apply H
Re-encode

$X^{a_1'} Z^{b_1'} H |0\rangle X^{a_2'} |0\rangle Z^{b_3'} |+\rangle$ →

$X^{a_1'} Z^{b_1'} H |0\rangle X^{a_2'} |0\rangle Z^{b_3'} |+\rangle$

⋮

← $X^{a_1'} Z^{b_1'} H |0\rangle X^{a_2'} |0\rangle Z^{b_3'} |+\rangle$

decode + check traps + measure output qubit

Transversal gate evaluation

- One-time pad allows transversal evaluation of Clifford gates

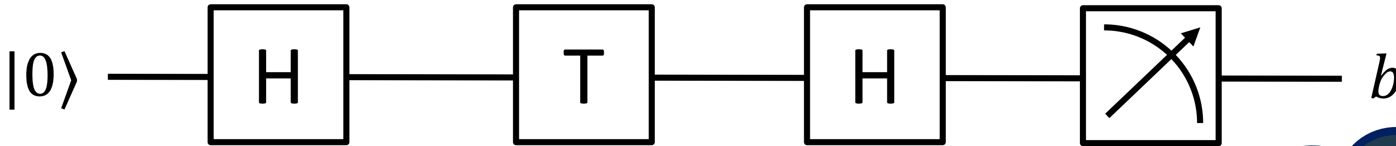
$$|\psi\rangle \quad \overset{H}{\mapsto} \quad H|\psi\rangle$$

$$Auth \downarrow \qquad\qquad \downarrow Auth$$

$$X^a Z^b |\psi\rangle X^{a'} |0\rangle Z^{b'} |+\rangle \quad \overset{HHH}{\mapsto} \quad \underbrace{Z^a X^b H|\psi\rangle Z^{a'} |+\rangle X^{b'} |1\rangle}_{Auth(H|\psi\rangle)}$$

- Clifford authentication allows transversal evaluation of Pauli gates

Running example



$(a_1, b_1), (a_2, b_2),$
$(a_3, b_3) \leftarrow_R \{0,1\}$

$X^{a_1}Z^{b_1}|0\rangle X^{a_2}|0\rangle Z^{b_3}|+\rangle$ →

$X^{a_1}Z^{b_1}|0\rangle X^{a_2}|0\rangle Z^{b_3}|+\rangle$

apply H →

$(a_1', b_1') \leftarrow (b_1, a_1)$
$(a_2', b_2') \leftarrow (b_2, a_2)$
$(a_3', b_3') \leftarrow (b_3, a_3)$

$$HX^{a_1}Z^{b_1}|0\rangle HX^{a_2}|0\rangle HZ^{b_3}|+\rangle$$
$$= X^{b_1}Z^{a_1}H|0\rangle Z^{a_2}H|0\rangle X^{b_3}H|+\rangle$$

measure X →

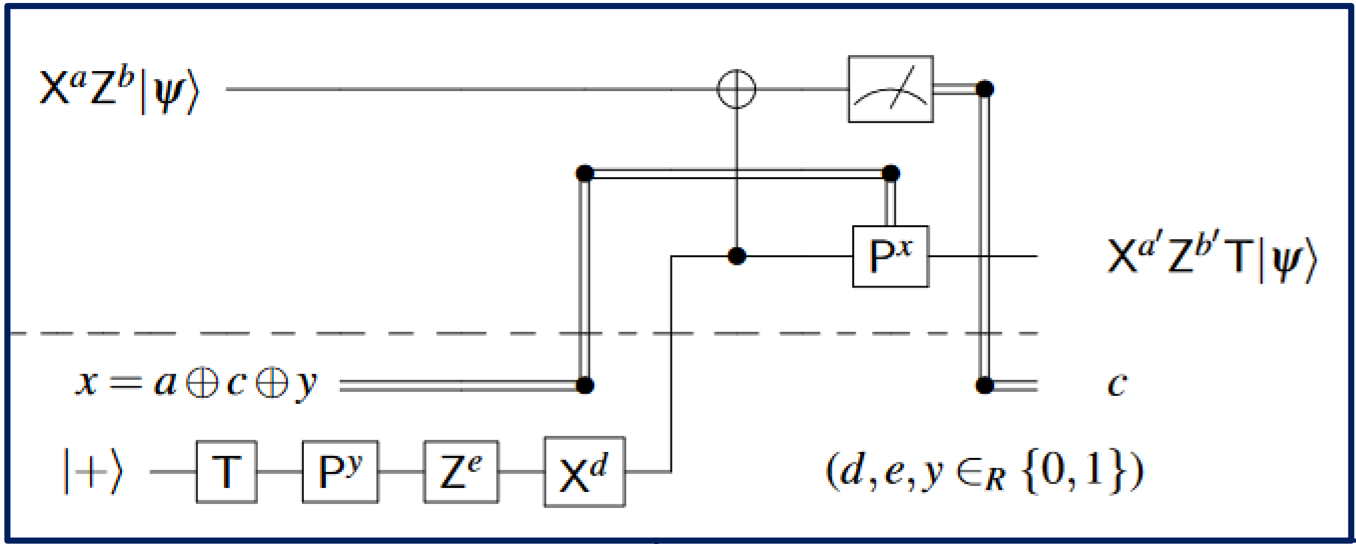
measure (X, X, X)

← (u_1, u_2, u_3)

check $u_3 = a_3'$
return $u_1 \oplus a_1'$

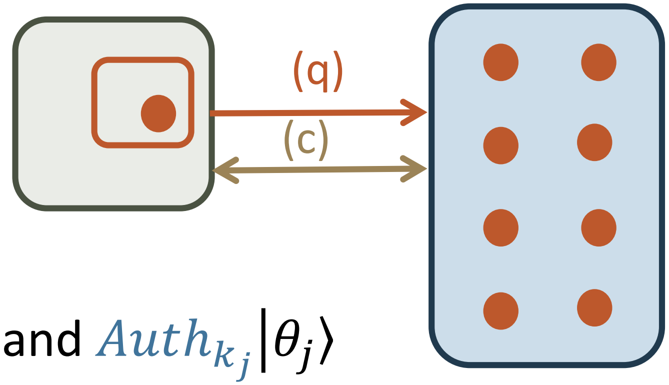
Transversal gate evaluation

- One-time pad allows transversal evaluation of Clifford gates
- Clifford authentication allows transversal evaluation of Pauli gates
- Polynomial-code authentication allows Clifford transversal gates
- Non-Clifford gates require magic states + classical communication



T-gate gadget: figure from [Broadbent'15]

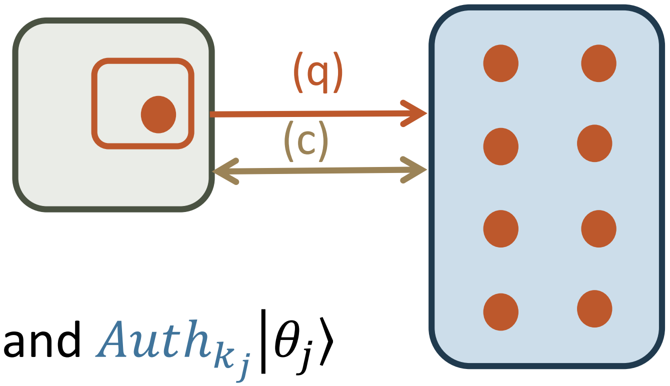
Verifiable blind delegated computation [ABOE'08,BFK'09]



- Verifier sends $Auth_{k_1}|C_1\rangle \otimes \cdots \otimes Auth_{k_n}|C_n\rangle$ and $Auth_{k_j}|\theta_j\rangle$
- To apply a Clifford gate:
  - Server applies gate transversally on authenticated qubits
  - Verifier updates authentication keys
- To apply non-Clifford gate:
  - Server uses authenticated magic state
  - Verifier and Server engage in protocol with classical communication
- Server measures output qubit and returns $(2k+1)$-bit outcome
  - Verifier checks traps and decodes final outcome

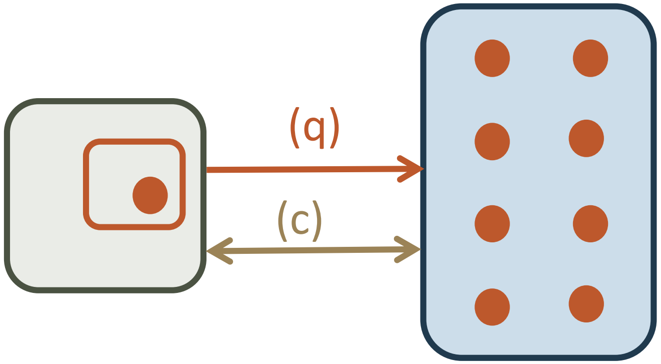
Verifiable blind delegated computation

[ABOE'08,BFK'09]



- Verifier sends $Auth_{k_1}|C_1\rangle \otimes \cdots \otimes Auth_{k_n}|C_n\rangle$ and $Auth_{k_j}|\theta_j\rangle$
- Blindness: authentication → one-time pad → perfect blindness
- Verifiability:
  - Arbitrary server = honest server + deviating unitary
  - Verifier's authentication + de-authentication induce Clifford twirl
  - Arbitrary attack reduced to random Pauli
  - Random Pauli likely to flip some traps
  - Intermediate classical communication rounds complicate analysis

Prepare & Send protocols: summary



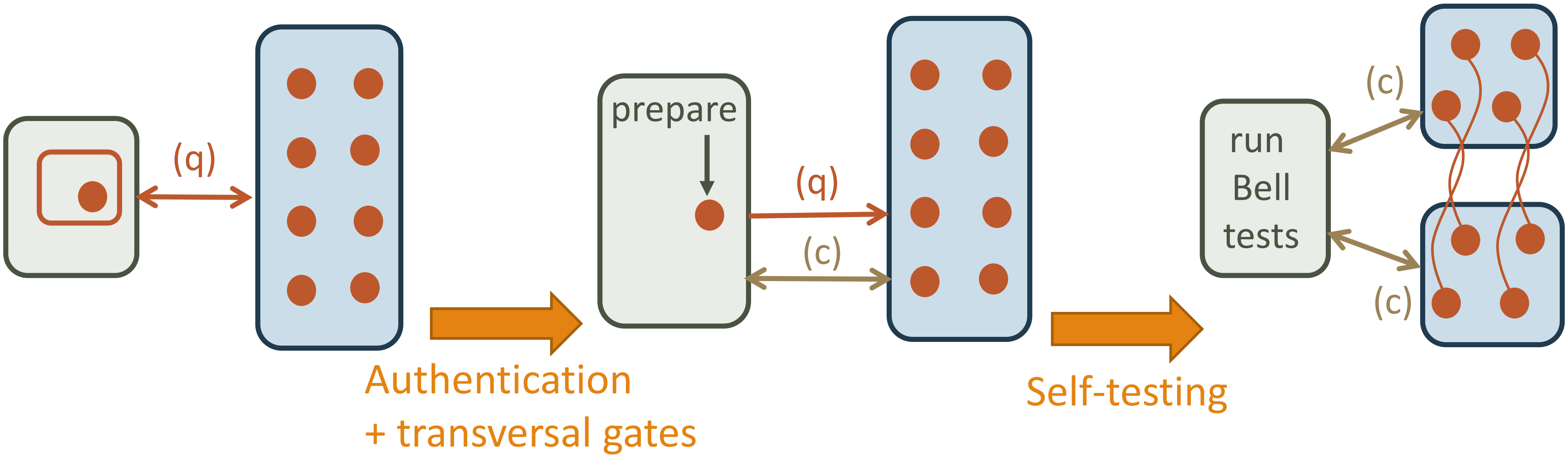
- One-way quantum communication + many-round classical communication
- [ADSS'17] quantum homomorphic computation with verification removes classical communication, under computational assumption

  Open: reduce interaction without making computational assumptions
- Verifier complexity:
  - [ABOE'08] (Circuit-based) Verifier needs $O(\log 1/\epsilon)$ qubits
  - [BFK'09] (Measurement-based) Verifier needs $O(1)$ qubits
- Protocols vulnerable to noise at the verifier

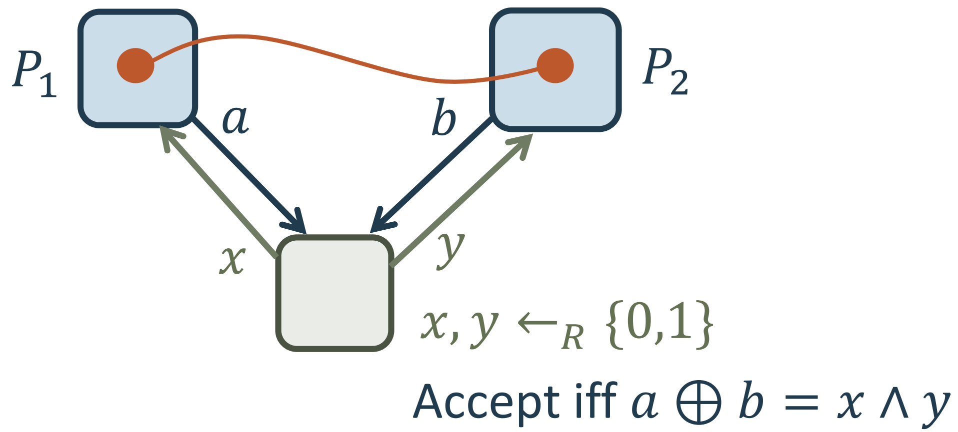
  Open: prepare & send fault-tolerant delegation

Part I(b): Two-prover delegation

Models for black-box verification



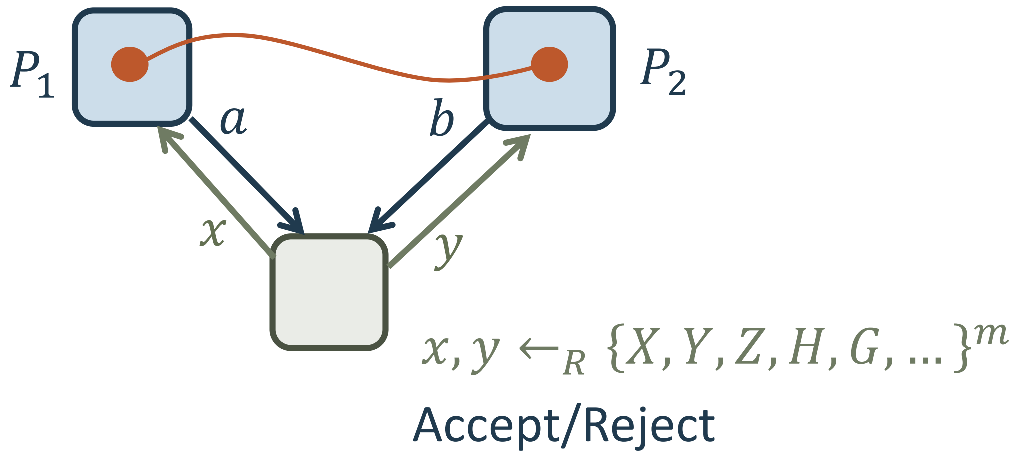
The CHSH game as a rigid self-test
[WS'88,MY'98,MYS'12,RUV'12]



- <u>Completeness</u>: Provers sharing an EPR pair succeed w.p. $\approx 85\%$
- <u>Soundness</u>: If provers succeed w.p. $\geq 85\% - \epsilon$, they must share an EPR pair, and P_1 measures in Pauli X $(x = 0)$ or Z $(x = 1)$ bases
- <u>Consequence</u>: After P_1 has returned a, P_2 has qubit in $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ which is known to V, but not to P_2

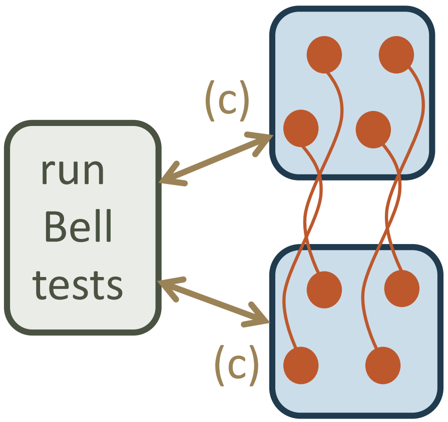
A rigid self-test for eigenstates of Clifford observables

[…,NV'17,CGJV'18]



- <u>Completeness</u>: Provers sharing m EPR pairs succeed w.p. 1
- <u>Soundness</u>: If provers succeed w.p. $\geq 1 - \epsilon$, they must share m EPR pairs, and P_1 measures i-th qubit using $A_i \in \{X, Y^*, Z, H^*, G^*, \dots\}$
- <u>Consequence</u>: After P_1 returns a, P_2 has m qubits in $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\theta\rangle, \dots\}$ which are known to V, but not to P_2

Two-prover verifiable delegation [RUV'12,CGJV'18]

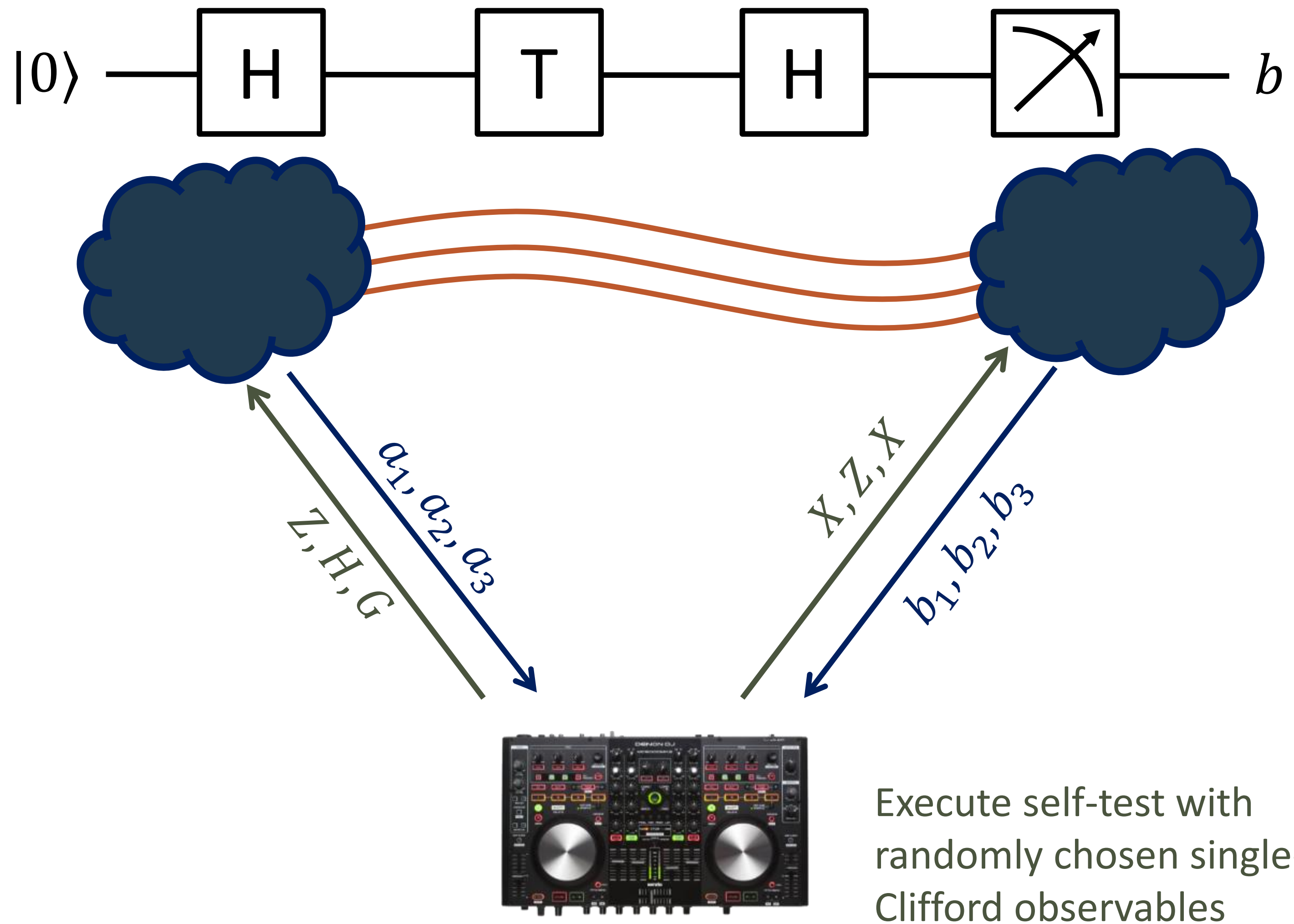


- w.p. ½: verifier executes self-test with provers
- w.p. ½:
  - Verifier instructs P_2 to make m-qubit measurement in randomly chosen bases
    → Given P_2's outcomes, P_1 has encrypted qubits

    $$X^a Z^b |\theta\rangle,\ \theta \in \left\{0,1,+,-,\frac{\pi}{4},\dots\right\}$$

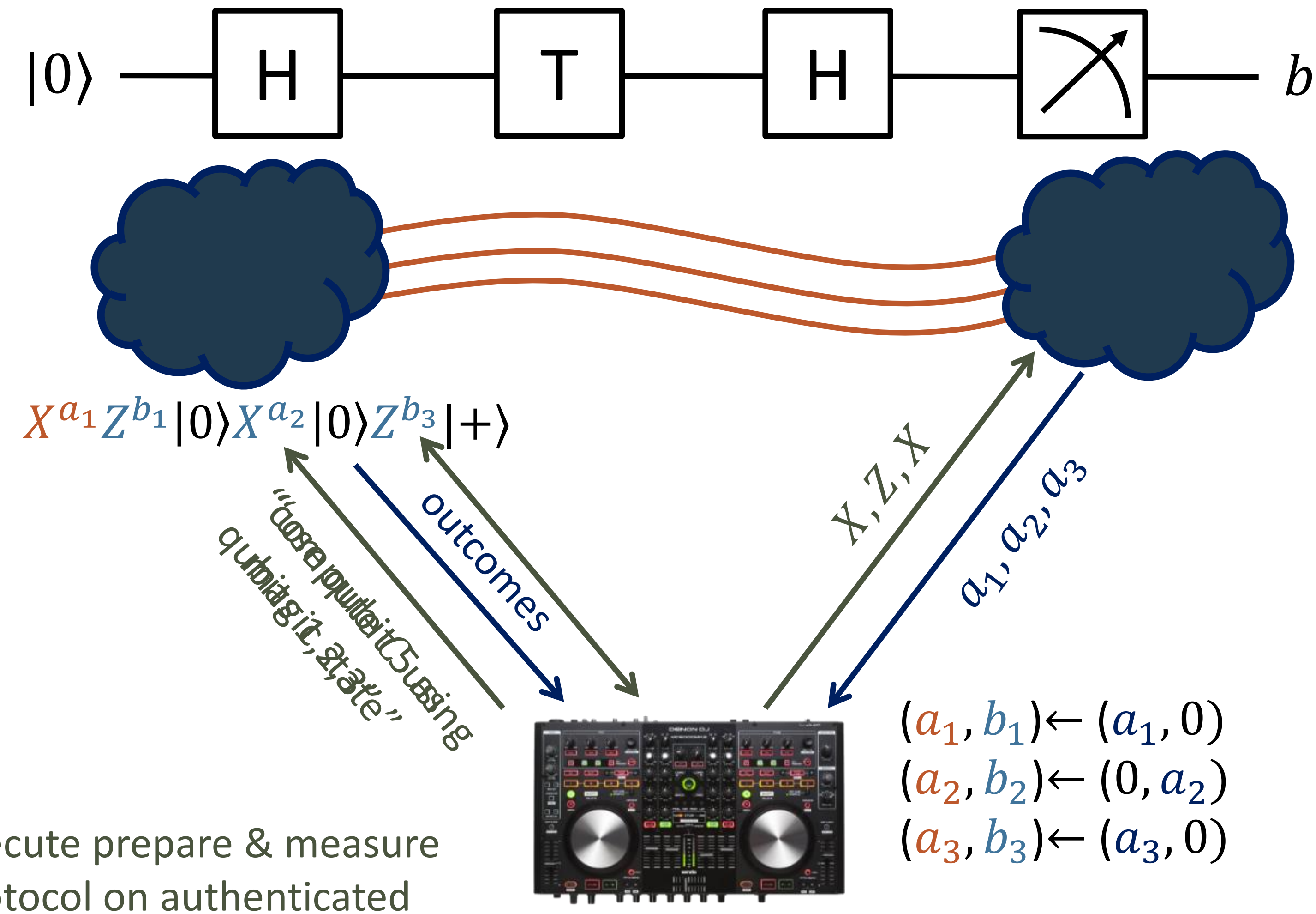
  - Verifier instructs P_1 to implement prepare & send protocol using designated qubits

Running example



Execute self-test with randomly chosen single-qubit Clifford observables

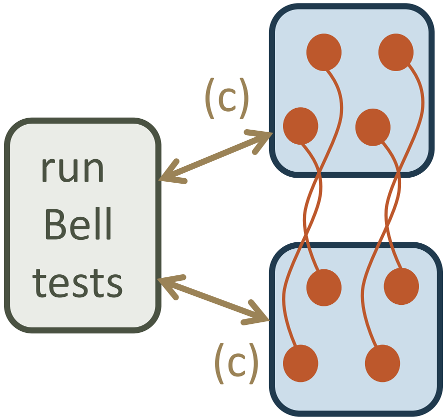
Running example



Execute prepare & measure protocol on authenticated qubits prepared by P_2

decode & check traps & return output

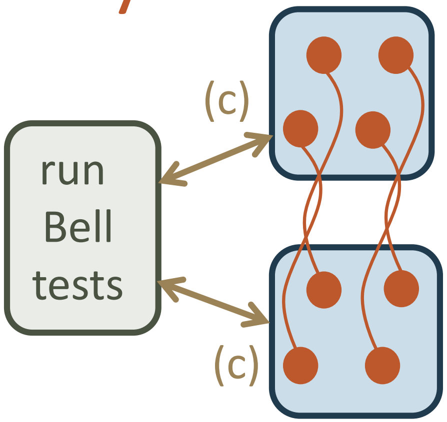
Two-prover verifiable delegation [RUV'12,CGJV'18]



- w.p. ½: verifier executes self-test with provers
- w.p. ½:
  - Verifier instructs P_1 to make m-qubit measurement in randomly chosen bases
  - Verifier instructs P_2 to perform implement prepare & send protocol using designated qubits

- <u>Blindness</u> follows from blindness for prepare & send, as long as provers do not communicate
- <u>Verifiability</u> follows from verifiability for prepare & send, additional $O(\epsilon^c)$ error from self-testing

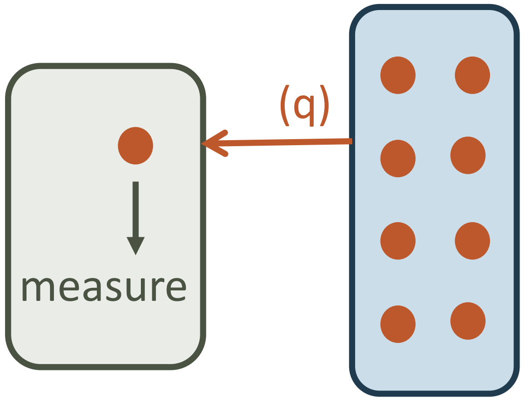
Two-prover protocols: summary



- Rigid self-tests allow preparation of eigenstates of single-qubit Clifford observables
  (partially) Open: non-Clifford eigenstates?
- Many-round classical interaction with two provers
- [Grilo'18] Single-round protocol in Hamiltonian model
  Protocol is not blind
  Open: single-round blind verifiable delegation protocol?
- Total communication ~linear in circuit size
  Open(?): sub-linear verifier? poly-logarithmic communication?
- Protocols extend to QMA verification if prover is given copies of QMA witness

Part II(a): Receive & Measure

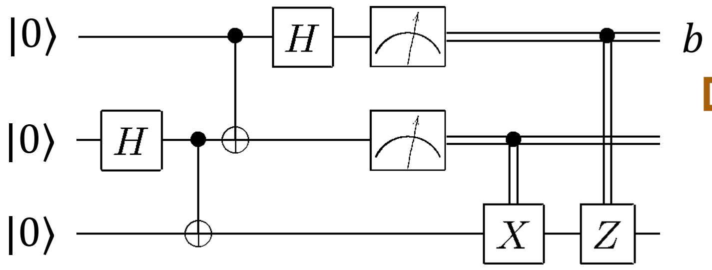
Receive & Measure protocols



- MBQC model:
  - Prover prepares resource state (e.g. cluster state)
  - Verifier either (i) checks stabilizers of resource state
    (ii) implements computation
  - Only needs single-qubit measurements in small number of bases
- Post-hoc model:
  - Prover prepares history state of Kitaev Hamiltonian associated with circuit
  - Verifier measures randomly chosen term in Hamiltonian
  - Only needs single-qubit measurements in two bases, but protocol not blind

Circuit-to-Hamiltonian

[Kitaev'99]



$$H = H_{in} + H_{clock} + H_{prop} + H_{out}$$

$$\Pr(C|0\rangle = 1) \geq 2/3 \implies \lambda_{min}(H) \leq a$$

$$\Pr(C|0\rangle = 1) \leq 1/3 \implies \lambda_{min}(H) \geq a + \delta$$

- Hamiltonian can be expressed in "XX/ZZ form":

  H is weighted sum of local terms of the form $X_i X_j$ or $Z_i Z_j$
- Gap δ scales as $1/|C|^2$
- Complexity of preparing ground state of H scales as complexity of C

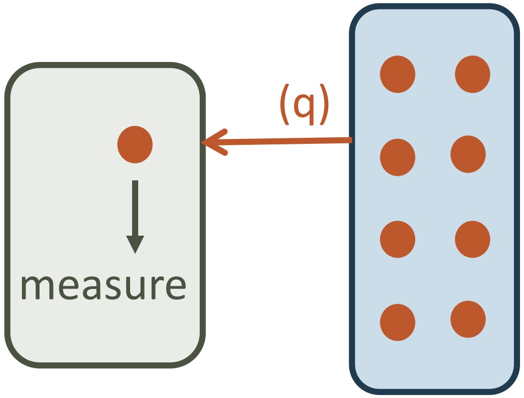
  (but may require higher depth)

Post-hoc verifiable delegation
[MF'16]

$$H = H_{in} + H_{clock} + H_{prop} + H_{out}$$

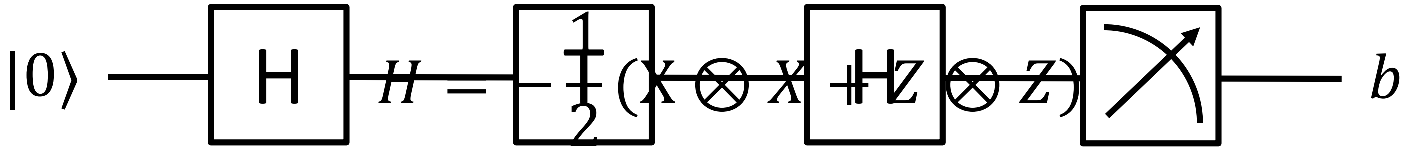
$$\Pr(C|0\rangle = 1) \geq 2/3 \implies \lambda_{min}(H) \leq a$$

$$\Pr(C|0\rangle = 1) \leq 1/3 \implies \lambda_{min}(H) \geq a + \delta$$



- Verifier computes H from C, sends to prover
- Prover prepares ground state of H
- Sends to verifier one qubit at a time
- Verifier secretly selects random local term $h_j = \mathrm{X}_{\mathrm{j}_1}\mathrm{X}_{\mathrm{j}_2}$ or $h_j = \mathrm{Z}_{\mathrm{j}_1}\mathrm{Z}_{\mathrm{j}_2}$
- Measures qubits j_1 and j_2 in required basis
- Repeat $1/\delta^2$ times to estimate energy

Running example



$$H = \frac{1}{2}(X \otimes X + Z \otimes Z)$$

H

flip coin $W \in \{X, Z\}$

prepare

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

first qubit

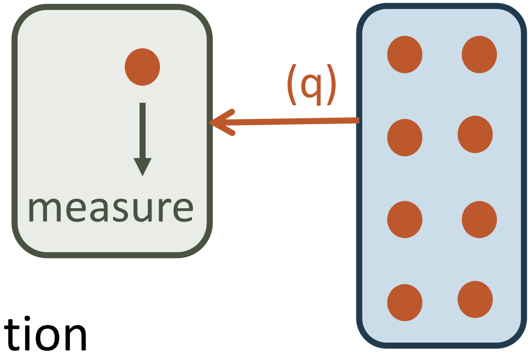
Measure in basis W $\rightarrow b_1$

second qubit

Measure in basis W $\rightarrow b_2$

Check: $b_1 b_2 = +1$

Receive & Measure protocols: summary



- One-way quantum communication
- Hamiltonian model requires repetition for gap amplification

  MBQC model requires repetition for resource state testing

  Total communication at least $\sim |C|^3$

  Open: protocol with linear communication complexity
- Blind protocols only in MBQC model
- Protocols vulnerable to noise at the verifier

  [GHK'18] give fault-tolerant protocol in Hamiltonian model; not blind

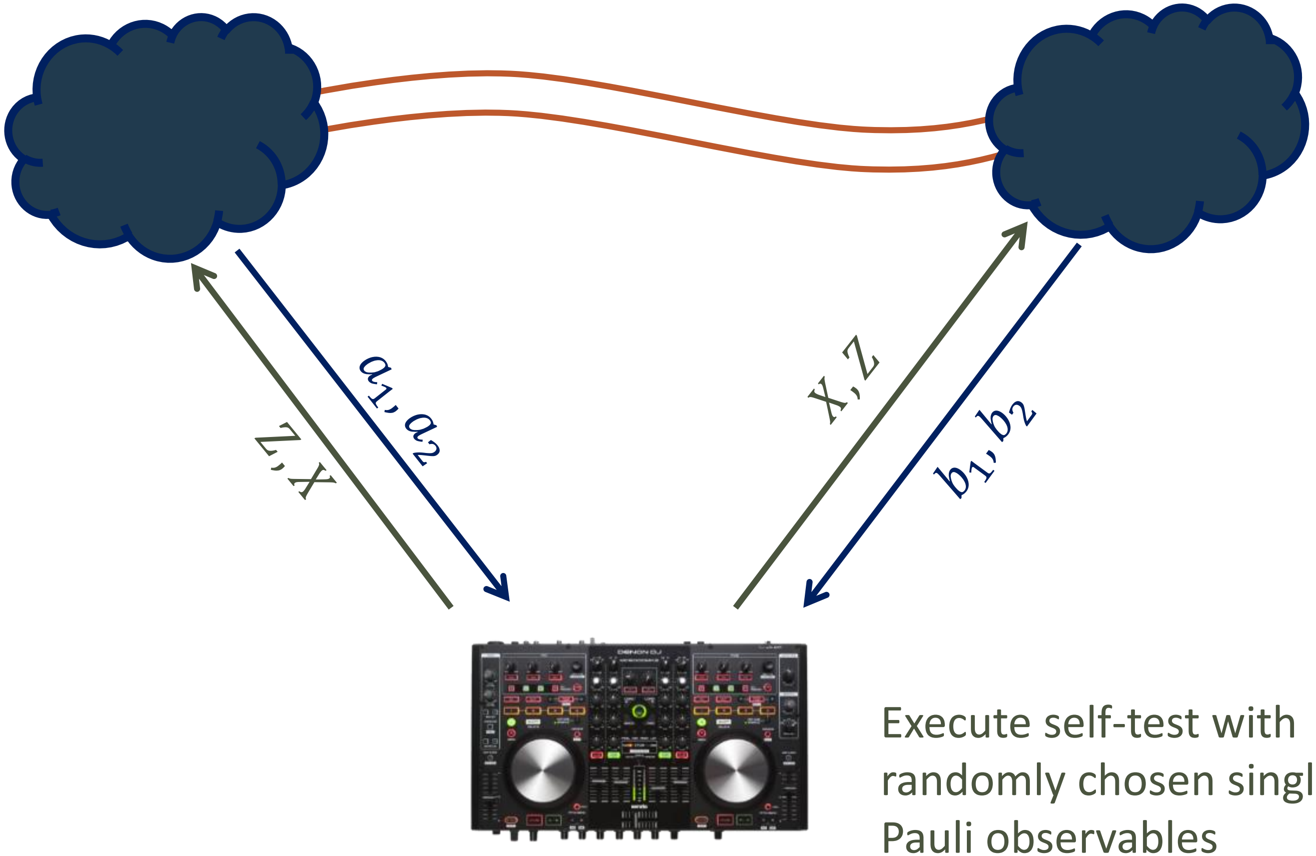
  Open: receive & measure fault-tolerant blind delegation

Part II(b):
Two-prover delegation

Running example

[Grilo'18]

$$H = -\frac{1}{2}(X \otimes X + Z \otimes Z)$$

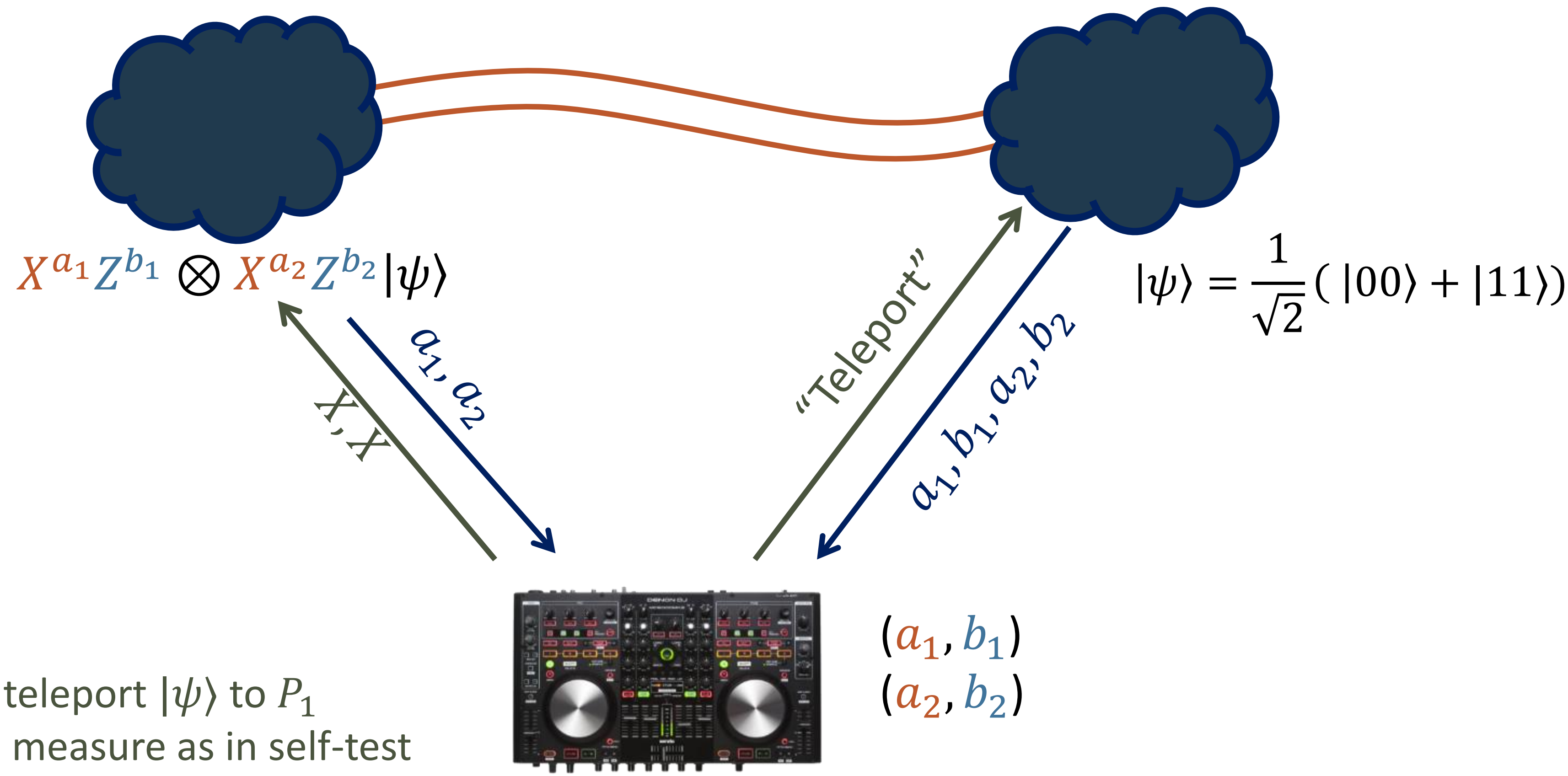


Execute self-test with randomly chosen single-qubit Pauli observables

Running example

[Grilo'18]

$$H = -\frac{1}{2}(X \otimes X + Z \otimes Z)$$



P_2: teleport $|\psi\rangle$ to P_1

P_1: measure as in self-test

Correct one-time pad and estimate energy

Part II(c): Commit & Reveal

Models for black-box verification

- Verifier "delegates" X and Z measurements to server
- Hurdle: Certify that reported measurement outcomes are obtained from a single underlying n-qubit state
- Idea: Use cryptography to "commit" prover to fixed n-qubit state

Committing to a bit

$b \in \{0,1\}$

$r \in_R \{0,1\}^n$

$c = com(b, r)$ ←

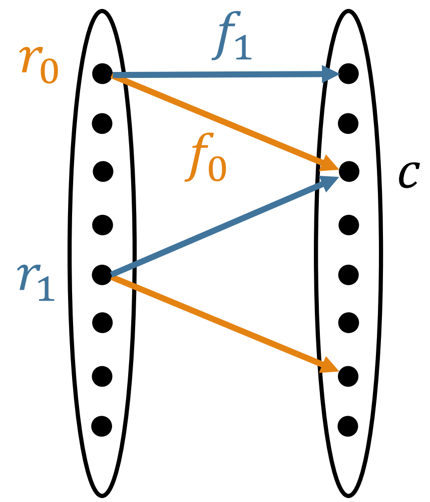
$d = reveal(b, r)$ ←

Return $(flag, b^*)$

- <u>Hiding</u>: c reveals no information about b $\quad c_{|b=0} \approx c_{|b=1}$

- <u>Binding</u>: For any efficient Bob, and any c such that $\Pr(flag = acc) \geq 0.01$, there is a b such that $\Pr(b^* = b \mid flag = acc) \geq 0.95$

Claw-free functions



$f_0, f_1 : \{0,1\}^n \to \{0,1\}^n$ a *claw-free* pair:

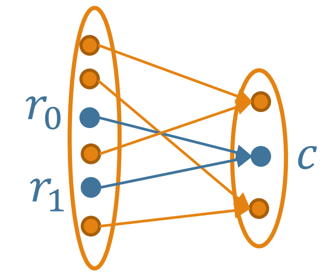
- Both f_0 and f_1 are bijections
- For every c in the range, there is a unique claw: a pair (r_0, r_1) such that $f_0(r_0) = f_1(r_1) = c$
- Claws are hard to find: no efficient procedure returns (r_0, r_1, c)

- Can construct based on “Learning with Errors” (LWE) problem
- f_0, f_1 are noisy multiplication by matrix A:

$$f_0(x) \approx A\,x + e, \quad f_1(x) \approx A(x - s) + e' \qquad \rightarrow \qquad r_1 \approx r_0 - s$$

Committing to a bit

$(f_0, f_1): \{0,1\}^n \rightarrow \{0,1\}^n$ a *claw-free* pair



$c = f_b(r)$

$d = (b, r)$

$b \in \{0,1\}$

$r \in_R \{0,1\}^n$

Check $f_b(r) = c$

Return b

- Perfectly hiding: Any c has exactly one preimage under each function
- Computationally binding:

If $\Pr(b^* = 0 | flag = acc) > 0.05$ and $\Pr(b^* = 1 | flag = acc) > 0.05$

then run Bob 100 times on c to find a claw

Committing to a *qubit*

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|R\rangle = \frac{1}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |r\rangle$$

$c = com(|\psi\rangle, |R\rangle)$ ←

$d_Z = Z\text{−}reveal(b, |R\rangle)$ ← Return $(flag, a_Z)$

$d_X = X\text{−}reveal(b, |R\rangle)$ ← Return $(flag, a_X)$

- Hiding: c reveals no information about $|\psi\rangle$
- Binding: For any efficient Bob and c such that $\Pr(flag = acc) \geq 0.01$ there is a ρ such that $a_Z \approx \mathrm{Tr}(Z\rho)$ and $a_X \approx \mathrm{Tr}(X\rho)$

Committing to a *qubit*

$$c = com(|\psi\rangle, |R\rangle)$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|R\rangle = \frac{1}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |r\rangle$$

$$|\psi\rangle \otimes |R\rangle \otimes |0^n\rangle \quad = \quad (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |r\rangle \otimes |0^n\rangle$$

$$\xrightarrow{\text{CTL-}f} \quad \frac{\alpha}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |0\rangle|r\rangle|f_0(r)\rangle + \frac{\beta}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |1\rangle|r\rangle|f_1(r)\rangle$$

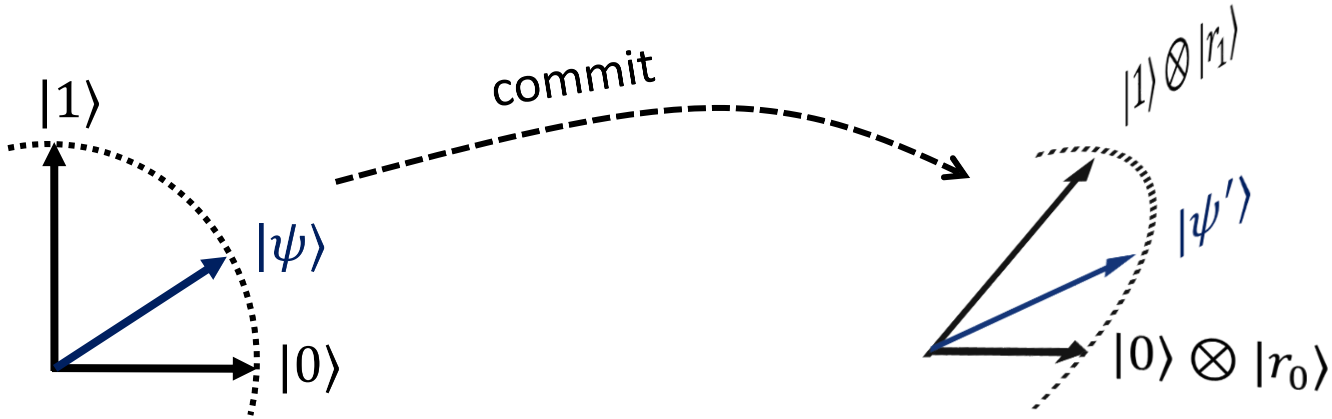
$$\xrightarrow{\text{meas. last register}} \quad (\alpha|0\rangle|r_0\rangle + \beta|1\rangle|r_1\rangle) \otimes |c\rangle$$

Committing to a *qubit*

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$c = com(|\psi\rangle, |R\rangle)$$

$$|R\rangle = \frac{1}{\sqrt{2^n}} \sum_{r \in \{0,1\}^n} |r\rangle$$



Committing to a *qubit*

$c = com(|\psi\rangle, |R\rangle)$

$d_Z = Z\text{-}reveal(b, |R\rangle)$

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$|R\rangle = \frac{1}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |r\rangle$

$$|\psi\rangle \otimes |R\rangle \otimes |0^n\rangle \quad = \quad (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |r\rangle \otimes |0^n\rangle$$

$$\xrightarrow{\text{CTL-}f} \quad \frac{\alpha}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |0\rangle|r\rangle|f_0(r)\rangle + \frac{\beta}{\sqrt{2^n}} \sum_{r\in\{0,1\}^n} |1\rangle|r\rangle|f_1(r)\rangle$$

$$\xrightarrow{\text{meas. last register}} \quad (\alpha|0\rangle|r_0\rangle + \beta|1\rangle|r_1\rangle) \otimes |c\rangle$$

- Hiding: c reveals no information about $|\psi\rangle$ ✔
- Z-reveal: Bob measures in computational basis and returns $d_Z = (b, r_b)$

  Alice checks $f_b(r_b) = c$ and returns "decoded bit" $a_Z = b$

Committing to a *qubit*

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$c = com(|\psi\rangle, |R\rangle)$

$d_X = \text{X}-reveal(b, |R\rangle)$

$$|R\rangle = \frac{1}{\sqrt{2^n}} \sum_{r \in \{0,1\}^n} |r\rangle$$

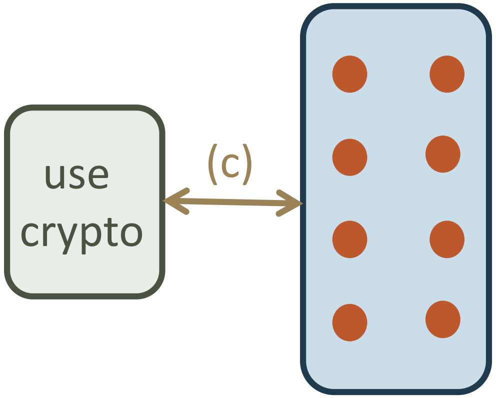
$$(\alpha|0\rangle|r_0\rangle + \beta|1\rangle|r_1\rangle) \xrightarrow{I \otimes H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{t \in \{0,1\}^n} (\alpha(-1)^{t \cdot r_0}|0\rangle + \beta(-1)^{t \cdot r_1}|1\rangle) \otimes |t\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{t \in \{0,1\}^n} (-1)^{t \cdot r_0} \quad Z^{t \cdot r_0 \oplus t \cdot r_1} \quad |\psi\rangle \otimes |t\rangle$$

- X-reveal: Bob measures in Hadamard basis and returns $d_X = (u, t)$

  Alice returns "decoded bit" $a_X = u \oplus (t \cdot r_0 \oplus t \cdot r_1)$

Commit & Reveal protocol
[Mahadev'18]



$$H = \sum_{(j_1,j_2)} \alpha_{j_1 j_2}(X_{j_1}X_{j_2} + Z_{j_1}Z_{j_2})$$

$$\Pr(C|0\rangle = 1) \geq 2/3 \implies \lambda_{min}(H) \leq a$$

$$\Pr(C|0\rangle = 1) \leq 1/3 \implies \lambda_{min}(H) \geq a + \delta$$

- Verifier computes H from C, sends to prover
- Prover prepares ground state of H

(same as post-hoc protocol)

- Prover individually commits to each qubit by sending $c_1, \ldots, c_n$
- Verifier secretly selects random local term $h_j = X_{j_1}X_{j_2}$ $(Z_{j_1}Z_{j_2})$
- Executes $X(Z)$-reveal phase with prover
- Records decoded outcomes $a_{X_{j_1}}a_{X_{j_2}}$ $(a_{Z_{j_1}}a_{Z_{j_2}})$
- Repeat $1/\delta^2$ times to estimate energy

Running example

$$H = -\frac{1}{2}(X \otimes X + Z \otimes Z)$$

H and f_0, f_1 and f'_0, f'_1 →

prepare $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

← commitments c, c'

run commitment procedure:

$$\frac{1}{\sqrt{2}}(|0, r_0\rangle|0, r'_0\rangle + |1, r_1\rangle|1, r'_1\rangle)$$

flip coin $W \in \{X, Z\}$

X-reveal? → (overlapping: Z)

Measure X (overlapping: Z)

← $b, r_b, b', r'_{b'}$

Check: $f_b(r_b) = c$

$f'_{b'}(r'_{b'}) = c'$

Record b, b'

← u, t, u', t'

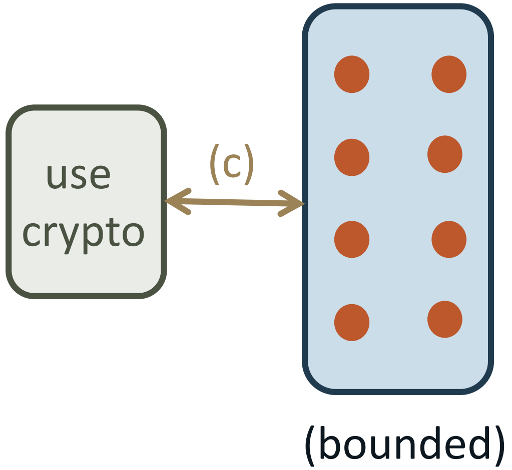
Set $a_X = u \oplus (t \cdot r_0 \oplus t \cdot r_1)$

$a'_X = u' \oplus (t' \cdot r'_0 \oplus t \cdot r'_1)$

Record $a_X a'_X$

Repeat $1/\delta^2$ times to estimate energy

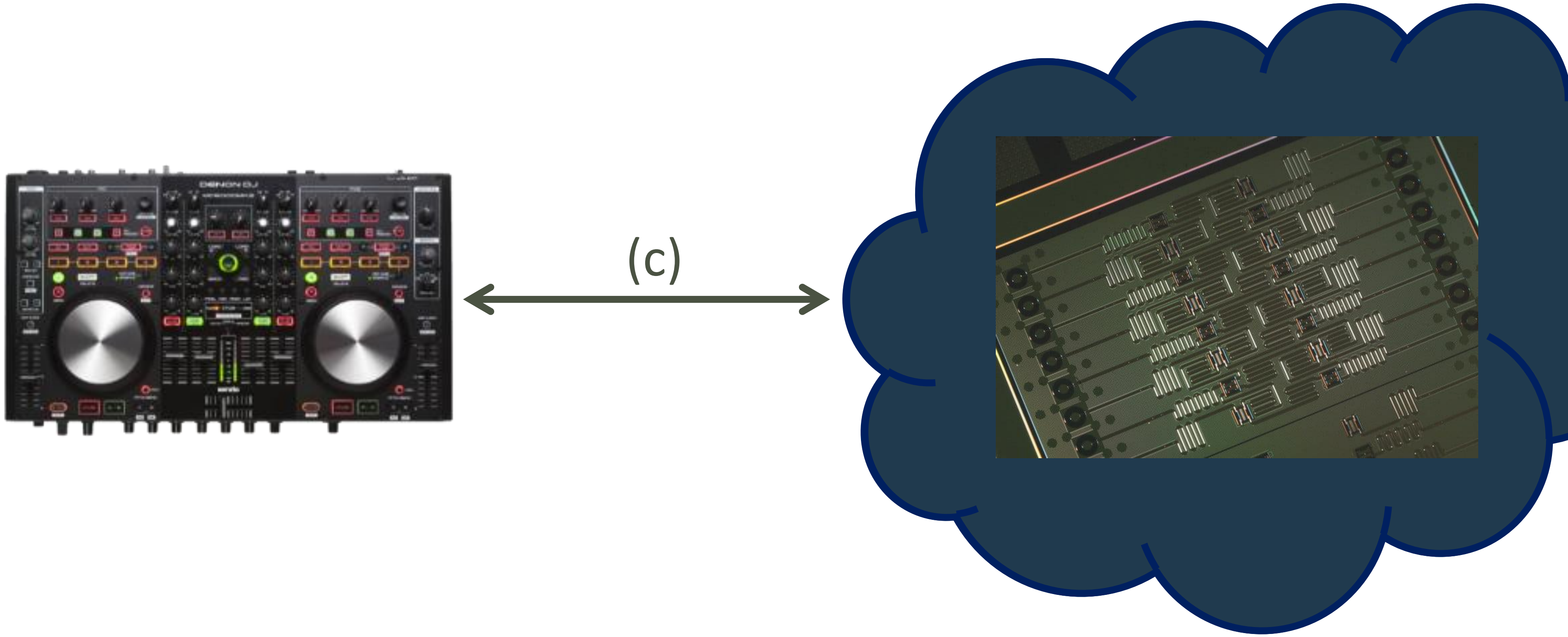
Commit & Reveal protocol: summary



- Hamiltonian model: protocol is not blind, but can be made blind by combining with quantum FHE
  Open: blind protocol in circuit or MBQC models?
- Complexity: cubic overhead due to Hamiltonian model
  Crypto overhead linear in security parameter
- Soundness guarantee: there *exists* a state that gives *computationally indistinguishable* measurement outcomes
  Open: computational assumption, information-theoretic guarantee?
- Claw-free function instantiated from learning with errors assumption (LWE)
  Open: more generic construction (e.g. quantum-secure OWF)?

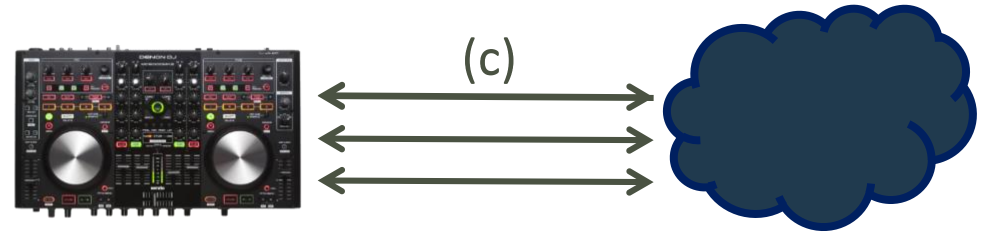
Coda:
An open question

An open question

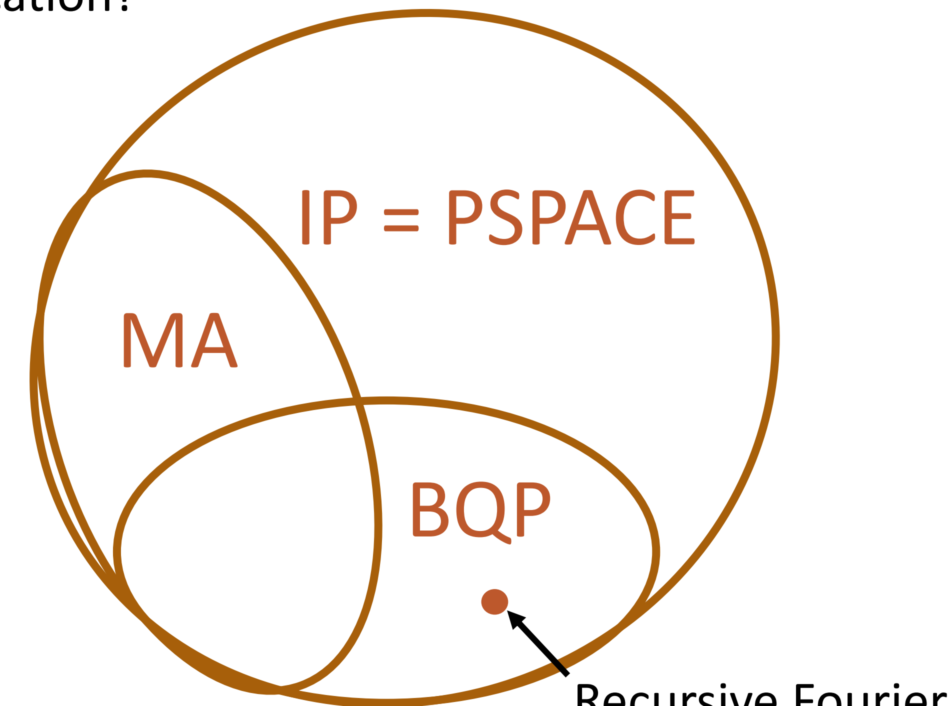


- Verifier is classical polynomial-time
- Communication channel is classical
- Verifier wants to determine $\Pr(C|0\rangle = 1)$

An open question



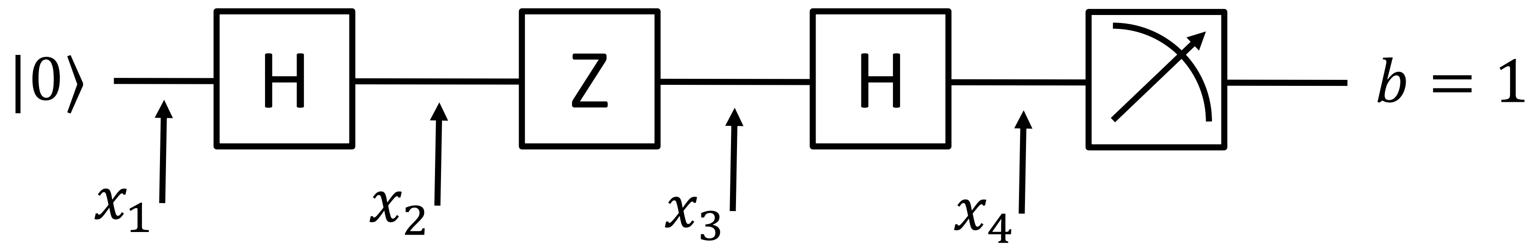
- Problems with efficient classical verification?
- MA = class of problems with efficient (probabilistic) verification
- Any problem in MA ∩ BQP has an efficiently verifiable solution
- Factoring, Graph Isomorphism



- IP = class of problems with efficient (probabilistic, interactive) verification
- IP Prover may not be efficient! Needs to compute exponentially large sums

Interactive proofs for BQP

- Feynman path integral: $\Pr(C|0\rangle = 1)$ is (square of) summation over exponentially many paths $$\sum_{path=(x_1,\ldots,x_T)} amplitude(x_1,\ldots,x_T)$$



- Amplitude of individual path is easy to compute

$$amplitude(0,1,1,0) = 1 \cdot \frac{1}{\sqrt{2}} \cdot (-1) \cdot \frac{1}{\sqrt{2}} = -\frac{1}{2}$$

- Amplitude is multilinear polynomial in $x_1, \ldots, x_T$

Interactive proofs for BQP

- Given $P \in \mathbb{F}_q[X_1, \dots, X_T]$ multilinear, compute $\sum_{x_1,\dots,x_T \in \{0,1\}} P(x_1, \dots, x_T)$

$S = \Sigma \; P(x_1, \dots, x_T)$

$p_T(z) = \Sigma \; P(x_1, \dots, x_{T-1}, z)$

$\Sigma_z p_T(z) = S \; ?$

$\widetilde{z_T}$

$\widetilde{z_T} \leftarrow_R \mathbb{F}_q$

$p_{T-1}(z) = \Sigma \; P(x_1, \dots, z, \widetilde{z_T})$

$\Sigma_z p_{T-1}(z) = p_T(\widetilde{z_T}) \; ?$

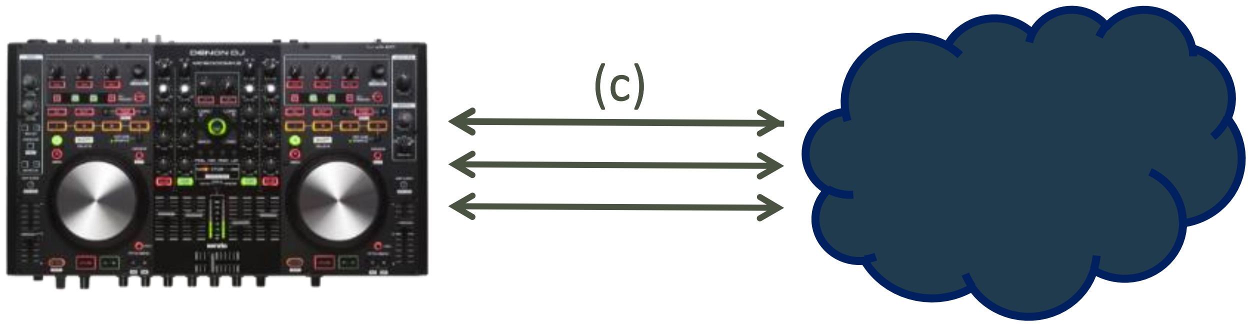
$\widetilde{z_{T-1}}$

$\widetilde{z_{T-1}} \leftarrow_R \mathbb{F}_q$

$p_0 = P(\widetilde{z_1}, \dots, \widetilde{z_T})$

$p_0 = P(\widetilde{z_1}, \dots, \widetilde{z_T}) \; ?$

Interactive proofs for BQP



- Any language in BQP has a classical-verifier interactive proof
- Prover needs to compute unphysical quantities
- Cannot be implemented using quantum computer
- [AG'17] give "quantum-inspired" variant of protocol
- Open: protocol with prover less powerful than PostBQP
- Challenge: allow prover to make statistical estimation errors

  while restricting capacity to cheat

Summary

Problem formulation

<u>Ideal functionality for verifiable & blind delegation</u>

circit C →

← bit b

$$b = \begin{cases} C(|0\rangle) & \text{if } e \in \{0,1\} \\ err & \text{if } e = 2 \end{cases}$$

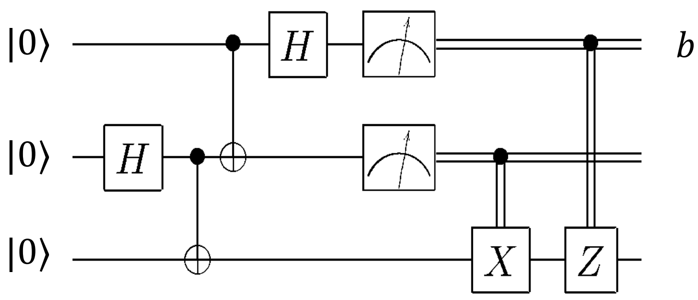
size T →

← trit e

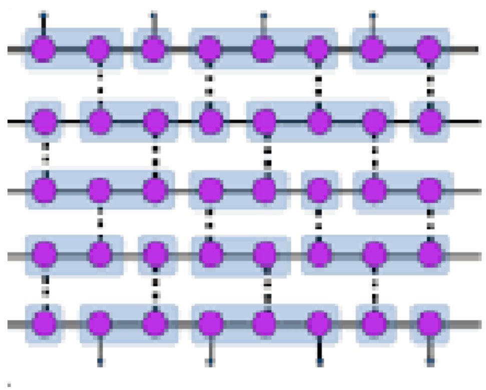
A protocol is verifiable & blind if no malicious party interacting with the honest party can distinguish from an interaction with the ideal functionality

Models of computation

Circuit model



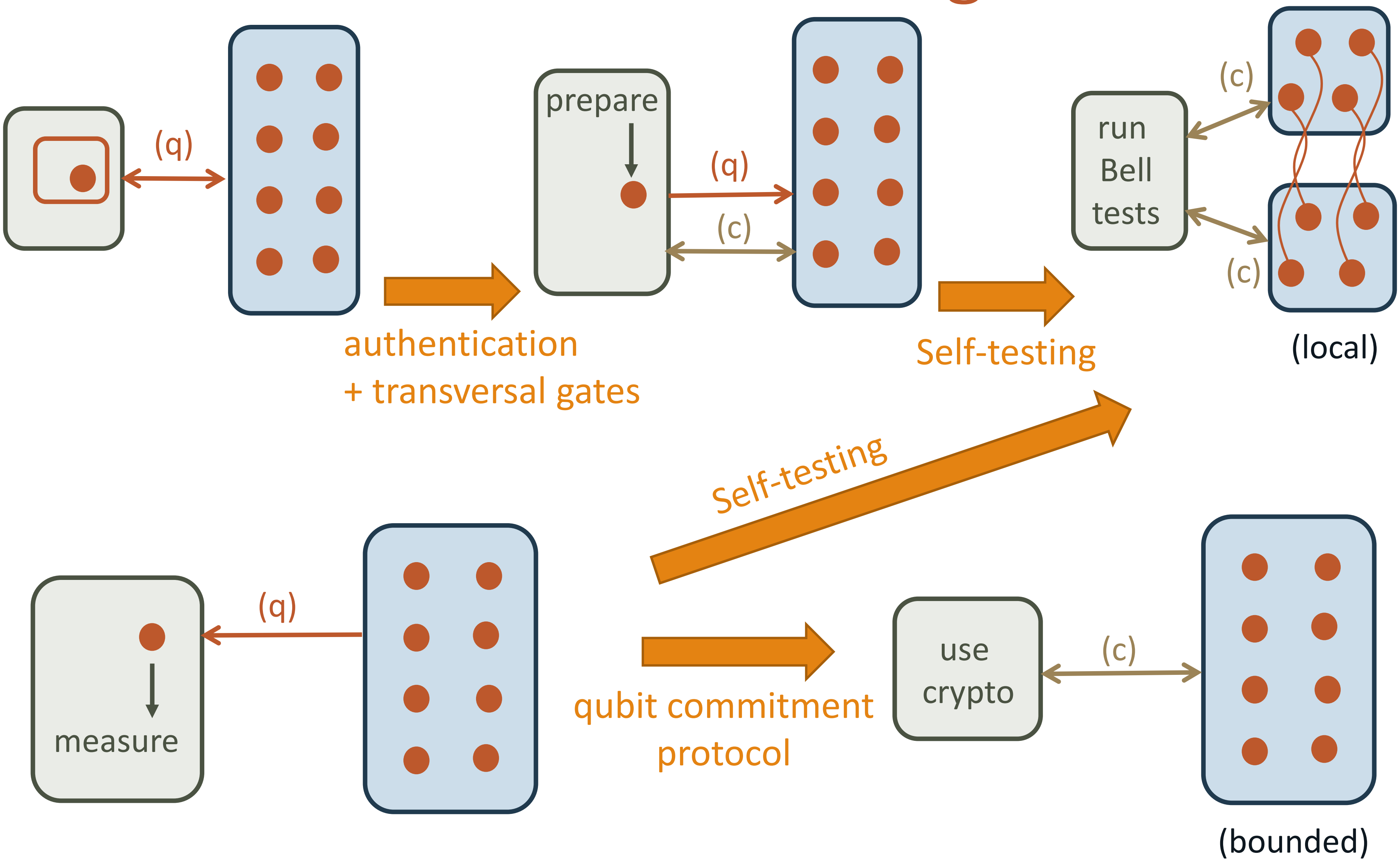
Measurement-based model



Hamiltonian model

$$H = H_{in} + H_{clock} + H_{prop} + H_{out}$$

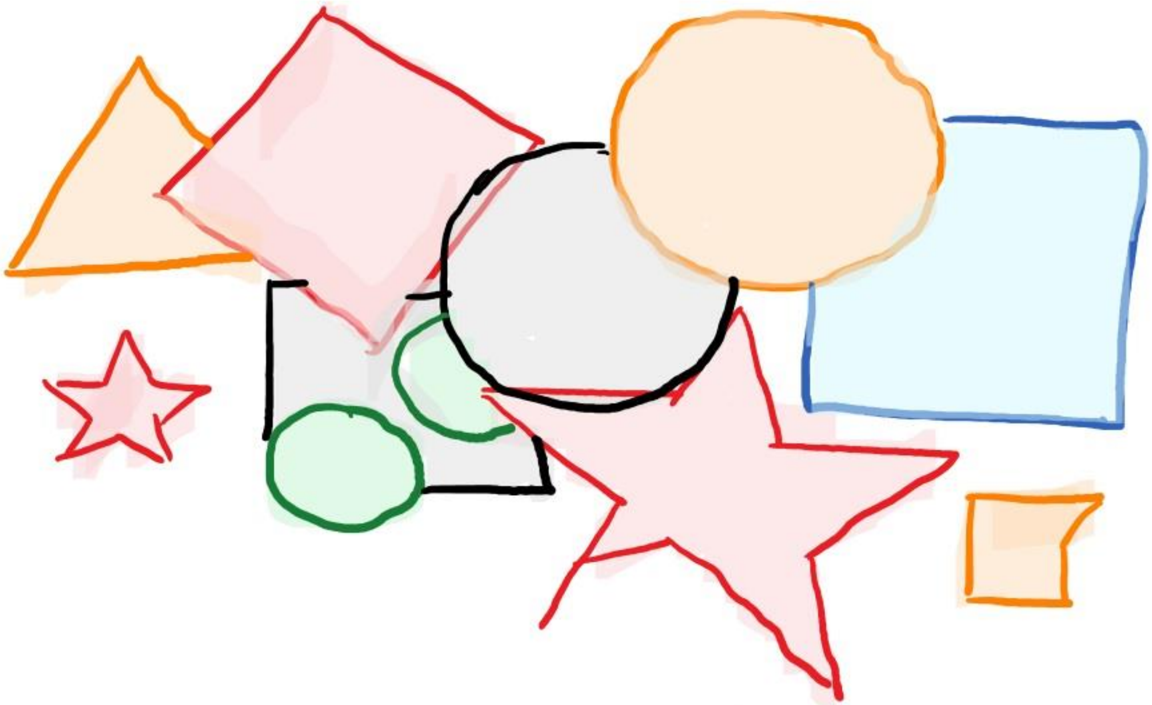
Protocols for verifiable delegation



Complexity considerations

Input: Circuit C, T gates, n qubits. eps: distance from ideal functionality

Protocol	Computation model	Verifier	Communication
Childs'05	Circuit	O(1)	O(T)
ABOE'08	Circuit	O(log 1/eps)	O(T log(1/eps))
BFK'09	MBQC	O(1)	O(T log(1/eps))
MF'13	MBQC	O(1)	O(T/eps^2)
MF'16	Hamiltonian	O(1)	O(T^3 log(1/eps))
CGJV'18	Circuit	classical	O(T/eps^c)
Mahadev'18	Hamiltonian	classical	O(T^3 log(1/eps)log(1/lambda))



Thank you

SLIDES:
HTTP://USERS.CMS.CALTECH.EDU/~VIDICK/VERIFICATION.{PPSX,PDF}

References

[ADSS'17] Alagic et al. "Quantum Fully Homomorphic Encryption With Verification." *arXiv:1708.09156*

[AG'17] Aharonov and Green. "A Quantum inspired proof of P^{\# P}\subseteq IP." *arXiv:1710.09078*

[BKB+'12] Barz et al. "Demonstration of blind quantum computing." *Science* 335.6066 (2012): 303-308.

[Broadbent'15] Broadbent. "How to verify a quantum computation." *arXiv:1509.09180*

[Childs'05] Childs. "Secure assisted quantum computation." *arXiv preprint quant-ph/0111046*

[DFPR'13] Dunjko et al. "Composable security of delegated quantum computation." *arXiv:1301.3662*

[GRB+'16] Greganti et al. "Demonstration of measurement-only blind quantum computing." *NJP* 18.1 (2016): 013020

[Grilo'18] Grilo. "Relativistic verifiable delegation of quantum computation." *arXiv:1711.09585*

[GHK'18] Gheorghiu et al. "A simple protocol for fault tolerant verification of quantum computation." *arXiv:1804.06105*

[GKK'17] Gheorghiu et al. "Verification of quantum computation: An overview of existing approaches." *arXiv:1709.06984*

[HZM+'17]Huang et al. "Experimental blind quantum computing for a classical client." *PRL* 119.5 (2017): 050503.

[Mahadev'18] Mahadev. "Classical verification of quantum computations." *arXiv:1804.01082*

[MF'13] Morimae and Fujii. "Blind quantum computation protocol in which Alice only makes measurements." *arXiv:1201.3966*

[MF'16] Morimae and Fitzsimons. "Post hoc verification with a single prover." *arXiv:1603.06046*

[MY'05] Mayers and Yao. "Self testing quantum apparatus." *quant-ph/0307205*

[MYS'12] McKague et al. "Robust Self Testing of the Singlet." *arXiv:1203.2976*.

[RUV'12] Reichardt et al. "A classical leash for a quantum system." *arXiv:1209.0448*.

[WS'88] Summers and Werner. "Maximal violation of Bell's inequalities for algebras of observables in tangent spacetime regions." *Annales de l'Institut Henri Poincare Physique Theorique*. Vol. 49. No. 2. 1988