Quantum Software Lab

Verification of Quantum Computation

An introduction to “Verifier-on-a-Leash: new schemes for verifiable delegated quantum computation with quasilinear resources”

PRESENTED BY
Maria Bahnă
María Gragera Garcés
David Mckey García
Mario Herrero González

Verifier-on-a-Leash: New Schemes for Verifiable Delegated Quantum Computation, with Quasilinear Resources

Andrea Coladangelo[1,2], Alex B. Grilo[3(✉)], Stacey Jeffery[3], and Thomas Vidick[1,2]

[1] Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA
[2] CMS, Caltech, Pasadena, USA
{acoladan,vidick}@cms.caltech.edu
[3] QuSoft and CWI, Amsterdam, The Netherlands
{alexg,jeffery}@cwi.nl

Outline

- Problem definition
- Different Schemes
- Key Properties
- Different Delegation Protocols
- Conclusions

Problem Definition, Motivation

Quantum Computers:

- Implementable + more computational power **Superiorita**
- Expensive
- Online Service (e.g IBM Cloud Service, IonQ, …)

→ **Guarantee server runs quantum computation.**

First protocol for verification with a classical client

Mahadev, U. (2018, October). Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 259-267). IEEE.

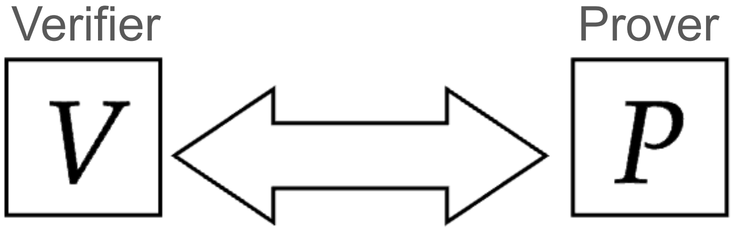
For blind verification with a quantum client

Fitzsimons, J. F., & Kashefi, E. (2017). Unconditionally verifiable blind quantum computation. Physical Review A, 96(1), 012303.

Problem Definition, Interactive proofs

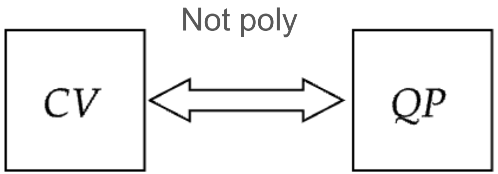
Goal: Interactive proof system for BQP where

- the verifier runs poly-time prob. computation
- an honest prover runs poly-time quantum computation
- the protocol is sound against any malicious prover
- additional property: the prover does not learn Q

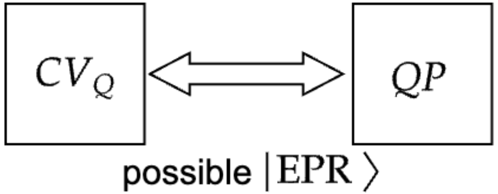


Different Schemes

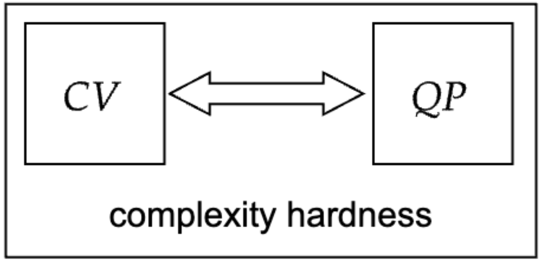
1.



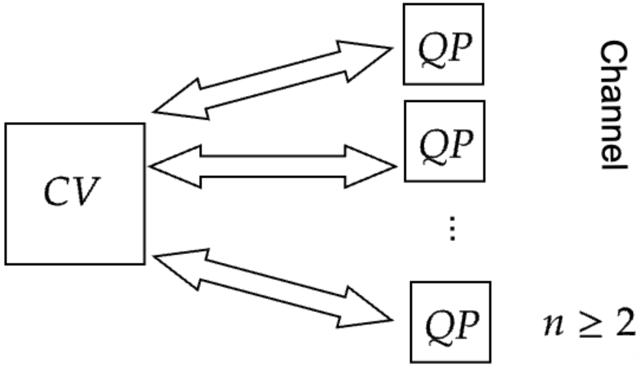
2.



3.



4.



The EPR Protocol, Scheme 2

- V wants to delegate a Circuit, *C* to P
- V has limited ‘quantum power’
- V wants to verify that P computed *C*
- V ‘switches’ between Computation & Test rouns
- These are indistinguishable to P
- P necessarily performs the same operations in both the Computation & Test rounds
- Use of encryption and ‘gate gadgets’

$$C\{X, Z, CNOT, H, T\}$$

$$|EPR\rangle = \frac{1}{\sqrt{2}}\left(|\psi\rangle|\psi\rangle + \left|\psi^{\perp}\right\rangle\left|\psi^{\perp}\right\rangle\right)$$

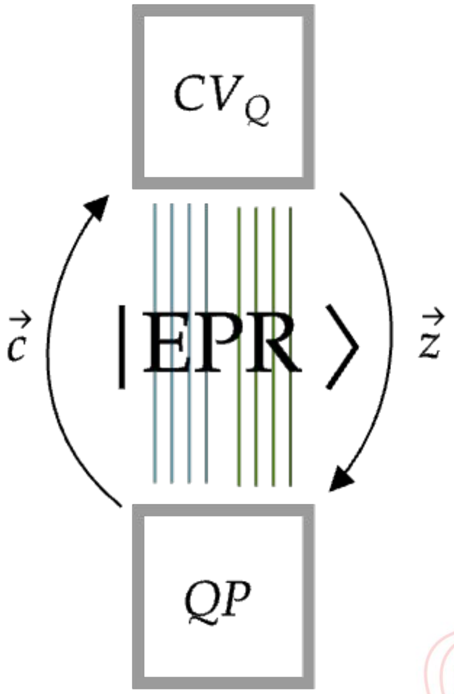
$$C\,|x\rangle$$

$$\mathbf{Id}\,|0\rangle$$

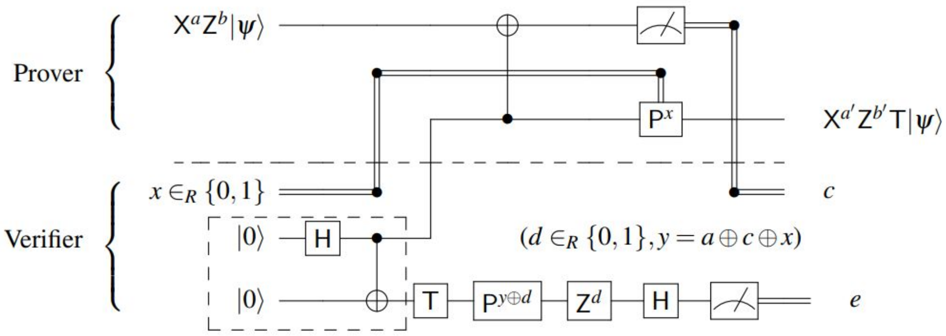
$$\mathbf{Id}\,|+\rangle$$

The EPR Protocol

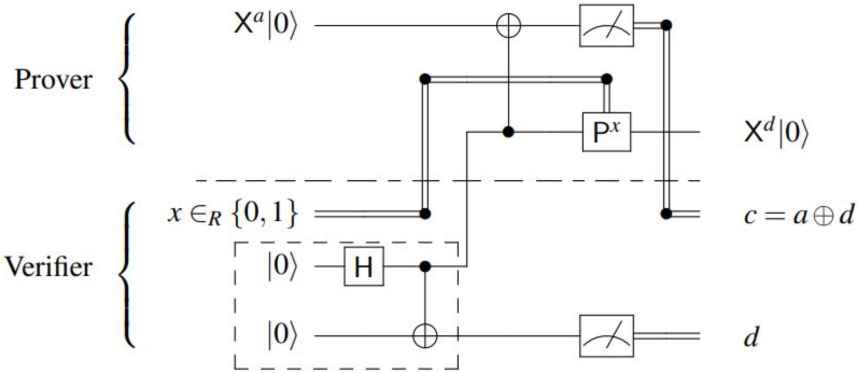
- CV and QP share EPR pairs
- CV sends $z_i \in \{0,1\}$
- QP sends back $c_i \in \{0,1\}$
- CV measures their half of EPR pairs which determines a Test vs Computation Round
- If QP passes all tests, then Accept
- Else Reject



T-gate Gadgets



- T gadget in a computational round
- Implements a T gate



- T gadget in a test round
- Implements the Identity

Broadbent, A. (2018). How to Verify a Quantum Computation. Theory OF Computing, 14(11), 1-37.

Key Properties

Completeness, there exists a prover (called an honest prover) such that the verifier accepts with probability $p \geq 2/3$.

Soundness, No malicious prover can convince V to accept with probability $p \geq 1/3$.

Blindness is a property of delegation protocols, which informally states that the prover learns nothing* about the verifier's circuit *C*.

Broadbent, A. (2018). How to Verify a Quantum Computation. Theory OF Computing, 14(11), 1-37.

Key Properties, Classical Game

$$\begin{array}{cc} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} & \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} \\ \begin{matrix} 0 & \quad 0 & \quad 0 \end{matrix} & \end{array}$$

$$b_{ij} \in \{0, 1\}$$

Key Properties, Classical Game

Prover 1

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Prover 2

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

$i \in \{1, 2, 3\}$

$[b'_{i1}, b'_{i2}, b'_{i3}]$

$j \in \{1, 2, 3\}$

$$\begin{bmatrix} b''_{1j} \\ b''_{2j} \\ b''_{3j} \end{bmatrix}$$

Verifier

Key Properties, Classical Game

Prover 1

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Best winning prob. for randomized strategies: 8/9

Prover 2

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

$[b'_{i1}, b'_{i2}, b'_{i3}]$

$i \in \{1,2,3\}$

Winning Conditions

$$b'_{i1} \oplus b'_{i2} \oplus b'_{i3} = 1$$
$$b''_{1j} \oplus b''_{2j} \oplus b''_{3j} = 0$$
$$b'_{ij} = b''_{ij}$$

$$\begin{bmatrix} b''_{1j} \\ b''_{2j} \\ b''_{3j} \end{bmatrix}$$

$j \in \{1,2,3\}$

Verifier

Key Properties, Quantum Game

$$\begin{array}{cc} \begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix} & \begin{array}{c} -I \otimes I \\ -I \otimes I \\ -I \otimes I \end{array} \\ \begin{array}{ccc} I \otimes I & \quad I \otimes I & \quad\quad I \otimes I \end{array} & \end{array}$$

Key Properties, Quantum Game

Prover 1

$$\begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix}$$

Prover 2

$$\begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix}$$

Please measure
I⊗Z X⊗I -X⊗Z

Winning Conditions

$$b'_{i1} \oplus b'_{i2} \oplus b'_{i3} = 1$$
$$b''_{1j} \oplus b''_{2j} \oplus b''_{3j} = 0$$
$$b'_{ij} = b''_{ij}$$

Please measure
I⊗Z Z⊗I Z⊗Z

Verifier

Key Properties, Rigidity

Questions Include the set of *m*-qubit Clifford observables
Answers: $\{0,1\}^m \times \{0,1\}^m$

Robust Rigidity Theorem: There is a constant *c*, and a function δ s.t.:
Completeness: If the players use strategy *S*, then they win *G* with prob. *c*.
Soundness: If the players use strategy *S'* that wins with probability *c* - ε, then $|S-S'| < \delta(\varepsilon)$.

Constant robustness: δ is constant in *m.*

A way to test **relations** between observables

What to consider when designing a protocol?

Provers
more or less

Blindness
reliability of the quantum device

Rounds
number of interactions to verify the quantum device

Resources
gate complexity of the provers, EPR pairs,number of bits

Different Delegation

	Provers	Rounds	Total Resources	Blind
RUV 2012 [RUV13]	2	$\text{poly}(n)$	$\geq g^{8192}$	yes
McKague 2013 [McK16]	$\text{poly}(n)$	$\text{poly}(n)$	$\geq 2^{153} g^{22}$	yes
GKW 2015 [GKW15]	2	$\text{poly}(n)$	$\geq g^{2048}$	yes
HDF 2015 [HPDF15]	$\text{poly}(n)$	$\text{poly}(n)$	$\Theta(g^4 \log g)$	yes
Verifier-on-a-Leash Protocol (Section 4)	2	$O(\text{depth})$	$\Theta(g \log g)$	yes
Dog-Walker Protocol (Section 5)	2	$O(1)$	$\Theta(g \log g)$	no

*n: number of qubits in the computation
*g: number of gates in the delegated circuit

Conclusion

- Motivation
  - Trust in our quantum computers
- Different Schemes
  - Key Properties: completeness, soundness & blindness
- Protocol Design
  - Provers, Rounds & Resources

Quantum Software Lab

Verifier-on-a Leash: New Schemes for Verifiable Delegated Quantum Computation with Quasilinear resources

Paper analysis

PRESENTED BY
Maria Bahnă
María Gragera Garcés
David Mckey García
Mario Herrero González

Verifier-on-a-Leash: New Schemes for Verifiable Delegated Quantum Computation, with Quasilinear Resources

Andrea Coladangelo[1,2], Alex B. Grilo[3(✉)], Stacey Jeffery[3], and Thomas Vidick[1,2]

[1] Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA
[2] CMS, Caltech, Pasadena, USA
{acoladan,vidick}@cms.caltech.edu
[3] QuSoft and CWI, Amsterdam, The Netherlands
{alexg,jeffery}@cwi.nl

Outline

- Recap Broadbent's protocol
- Introduction to Verifier-on-a Leash
- Non-local games & Rigidity
- Scalability
- Blindness
- Conclusion

Recap: Broadbent's Protocol

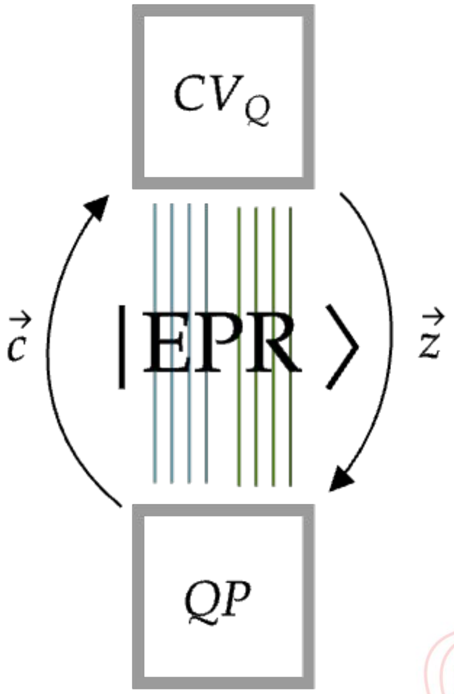
- V wants to delegate a Circuit, *C* to P
- V has limited 'quantum power'
- V wants to verify that P computed *C*
- V 'switches' between Computation & Test rounds
- These are indistinguishable to P
- P necessarily performs the same operations in both the Computation & Test rounds

$$C\{X, Z, CNOT, H, T\}$$

$$|EPR\rangle = \frac{1}{\sqrt{2}}\left(|\psi\rangle|\psi\rangle + \left|\psi^{\perp}\right\rangle\left|\psi^{\perp}\right\rangle\right)$$

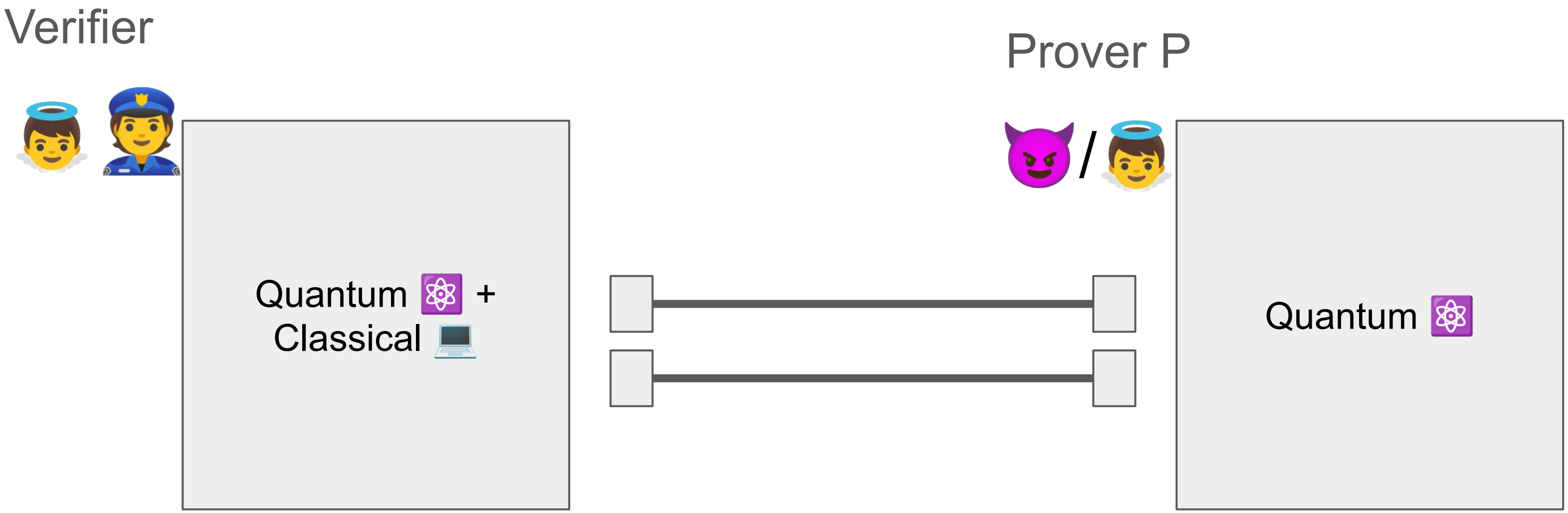
Recap: Broadbent's Protocol

- *CV* and *QP* share EPR pairs
- *CV* sends $z_i \in \{0,1\}$
- *QP* sends back $c_i \in \{0,1\}$
- CV measures their half of EPR pairs which determines a Test vs Computation Round
- If *QP* passes all tests, then Accept
- Else Reject



The Previous Scheme

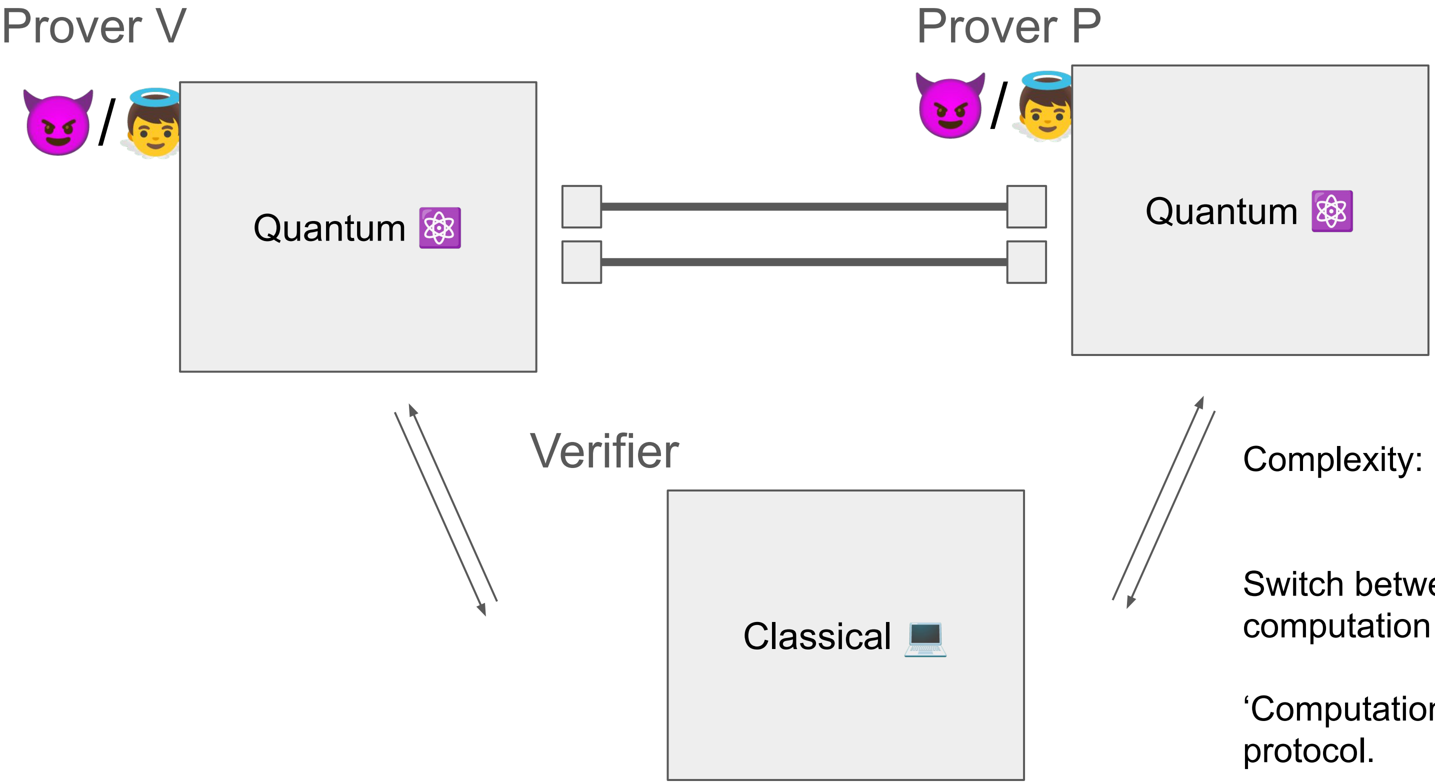
Verifier is trusted



Pro: Efficient Protocol O(g)
Con: requires verifier to have ‘quantum power’

The New Scheme: Verified on a Leash

Delegated Verifier is NOT trusted



Complexity: $O(g \log g)$

Switch between test and computation rounds.

‘Computation round’ = broadbent protocol.

Different Cases

Reduction := Use the same assumptions of rigidity & robustness as case 2.

	Prover V	Prover P	Argument
1.	😇	😇	Broadbent scenario (no problem) Correctness case
2.	😇	😈	Broadbent scenario (no problem) Security Broadbent scheme
3.	😈	😇	Clifford rigidity game (new) Rigidity game → reduction to 2.
4.	😈	😈	Clifford rigidity game (new) Rigidity game → reduction to 2.

Recap: Foundations

→ **Guarantee server runs quantum computation.**

Blindness: is a property of delegation protocols, which informally states that the prover learns nothing* about the verifier's private circuit.

Robust Rigidity Theorem: There is a constant *c*, and a function δ s.t.:
Completeness: If the players use strategy *S*, then they win *G* with prob. *c*.
Soundness: If the players use strategy *S'* that wins with probability *c* - ε ,
then |*S*-*S'*| < δ(ε).

Constant robustness: δ is constant in *m.*

$$\begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix}$$

A way to test **relations** between observables

Recap: Non-local games

Prover 1

$$\begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix}$$

Prover 2

$$\begin{bmatrix} I \otimes Z & X \otimes I & -X \otimes Z \\ Z \otimes I & I \otimes X & -Z \otimes X \\ Z \otimes Z & X \otimes X & -(XZ) \otimes (XZ) \end{bmatrix}$$

Please measure
I⊗Z X⊗I -X⊗Z

Winning Conditions

$$b'_{i1} \oplus b'_{i2} \oplus b'_{i3} = 1$$
$$b''_{1j} \oplus b''_{2j} \oplus b''_{3j} = 0$$
$$b'_{ij} = b''_{ij}$$

Please measure
I⊗Z Z⊗I Z⊗Z

Verifier

Robustness under Clifford rigidity game

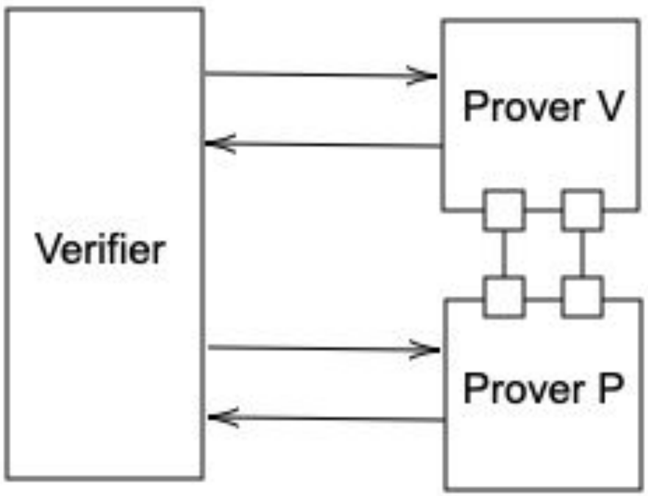
1) The verifier asks both PV and PP to measure specific tensor-product observables:

$$I \otimes Z \qquad\qquad -X \otimes Z$$

2) Expected measurement outcomes depend on the shared EPR state between PV and PP. Each prover measures half of the EPR pair, and results should obey Clifford algebra relations.

$$\langle (X \otimes I)(I \otimes Z) \rangle = \langle X \otimes Z \rangle$$

3) Broadbent logic => verification under thruthfull PV.



	Prover V	Prover P	Argument
1.	👼	👼	Broadbent scenario (no problem) Correctness case
2.	👼	😈	Broadbent scenario (no problem) Security Broadbent scheme

Robustness under Clifford rigidity game

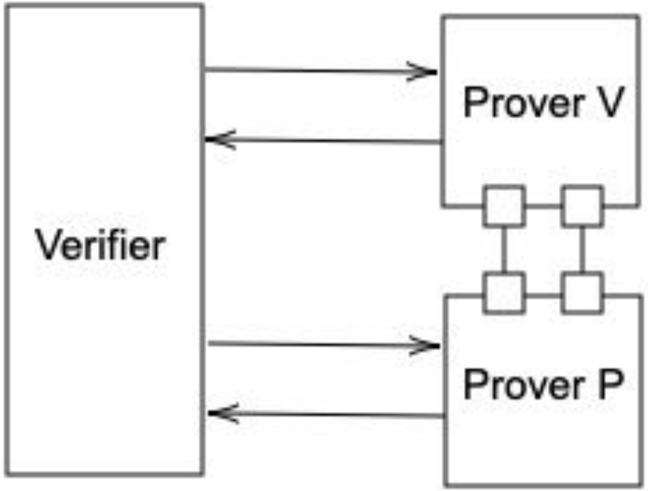
1) The verifier asks both PV and PP to measure specific tensor-product observables:

$$I \otimes Z \qquad\qquad - X \otimes Z$$

2) Expected measurement outcomes depend on the shared EPR state between PV and PP. Each prover measures half of the EPR pair, and results should obey Clifford algebra relations.

$$\langle (X \otimes I)(I \otimes Z) \rangle = \langle X \otimes Z \rangle$$

3) Lying PV???!!!



Self tests

Self tests: states

$$\underbrace{\overbrace{\left\|(V_A \otimes V_B)|\psi\rangle_{AB}}} - \underbrace{\overbrace{|\mathrm{EPR}\rangle^{\otimes m}_{A'B'}|\mathrm{AUX}\rangle_{\hat{A}\hat{B}}\right\|^2}} = \underbrace{\overbrace{O(\sqrt{\varepsilon})}},$$

Behaviour of our system

Expected output (for non-malicious players)

Scaling of distance between the states

Self-testing certifies that the shared resources are the correct EPR pairs if the players successfully execute CLIFF(Σ, m). Robustness ensures that the shared pairs are close to correct if they win with a probability of at least $1-\varepsilon$.

Self tests: operations

$$\underbrace{\mathop{\mathrm{E}}_{W\in\Sigma^m,\, c\in\{0,1\}^m} \left\| \mathrm{Id}_A \otimes \left(\underbrace{V_B W(c)}_{\text{Requested operations}} - \underbrace{\tau_W(c) V_B}_{\text{Performed operations}}\right) |\psi\rangle_{AB} \right\|^2}_{\text{Expectation values}} = \underbrace{O(\mathrm{poly}(\varepsilon))}_{\text{Distance between operations acting on states}}$$

Self-testing certifies that the shared resources are the correct operations if the players successfully execute CLIFF(Σ, m). Robustness ensures that the shared pairs are close to correct if they win with a probability of at least $1-\varepsilon$.

Delegation Protocols with Classical

"Total Resources" refers to the gate complexity of the provers, the number of EPR pairs of entanglement needed, and the number of bits of communication in the protocol

	Provers	Rounds	Total Resources	Blind
RUV 2012 [RUV13]	2	poly(n)	$\geq g^{8192}$	yes
McKague 2013 [McK16]	poly(n)	poly(n)	$\geq 2^{153} g^{22}$	yes
GKW 2015 [GKW15]	2	poly(n)	$\geq g^{2048}$	yes
HDF 2015 [HPDF15]	poly(n)	poly(n)	$\Theta(g^4 \log g)$	yes
Verifier-on-a-Leash Protocol (Section 4)	2	O(depth)	$\Theta(g \log g)$	yes
Dog-Walker Protocol (Section 5)	2	$O(1)$	$\Theta(g \log g)$	no

*n: number of qubits in the computation
*g: number of gates in the delegated circuit

Scalability: Where does the glog(g) come from?

Asymptotic complexity comes from:

- Universal circuit
- Sampling (rigidity test: provers implement correct ops on correct resource state)

	Provers	Rounds	Total Resources	Blind
RUV 2012 [RUV13]	2	poly(n)	$\geq g^{8192}$	yes
McKague 2013 [McK16]	poly(n)	poly(n)	$\geq 2^{153} g^{22}$	yes
GKW 2015 [GKW15]	2	poly(n)	$\geq g^{2048}$	yes
HDF 2015 [HPDF15]	poly(n)	poly(n)	$\Theta(g^4 \log g)$	yes
Verifier-on-a-Leash Protocol (Section 4)	2	O(depth)	$\Theta(g \log g)$	yes
Dog-Walker Protocol (Section 5)	2	$O(1)$	$\Theta(g \log g)$	no

Scalability: Why doesn't it grow higher?

Distance between states

$$\overbrace{\underbrace{\left\| (V_A \otimes V_B)|\psi\rangle_{AB}}_{\text{Behaviour of our system}}} - \overbrace{\underbrace{|\mathrm{EPR}\rangle^{\otimes m}_{A'B'}|\mathrm{AUX}\rangle_{\hat{A}\hat{B}}\right\|^2}_{\text{Expected output (for non-malicious players)}}} = \overbrace{\underbrace{O(\sqrt{\varepsilon})}_{\text{Scaling of distance between the states}}},$$

Amount of tests (self testing EPR pairs) you need to perform is not related to the number of EPR pairs (and thus number of gates) => amount of effort to certify the victory of the game is independent of "size".

Scalability: Why doesn't it grow higher?

Distance between operations

$$\underbrace{\mathop{\mathrm{E}}_{W\in\Sigma^m,\, c\in\{0,1\}^m} \left\| \mathrm{Id}_A \otimes \big(\underbrace{V_B W(c)}_{\text{Requested operations}} - \underbrace{\tau_W(c) V_B}_{\text{Performed operations}}\big) |\psi\rangle_{AB} \right\|^2}_{\text{Expectation values}} = \underbrace{O(\mathrm{poly}(\varepsilon))}_{\text{Distance between operations acting on states}}$$

HS-norm allows for composability

Scalability: Why doesn't it grow higher?

Distance between operations applied on a resource

$$\mathop{\mathbb{E}}_{W\in\Sigma^m} \sum_{u\in\{\pm 1\}^m} \Big\| \overbrace{\underbrace{V_A \mathrm{Tr}_B\big((\mathrm{Id}_A \otimes W_B^u)|\psi\rangle\langle\psi|_{AB}(\mathrm{Id}_A \otimes W_B^u)^\dagger\big) V_A^\dagger}_{\text{Honest operations applied on resource}}} - \overbrace{\underbrace{\sum_{\lambda\in\{\pm\}} \Big(\bigotimes_{i=1}^m \frac{\sigma_{W_{i,\lambda}}^{u_i}}{2}\Big) \otimes \tau_\lambda}_{\text{Performed operations}}} \Big\|_1$$

Honest operations applied on resource

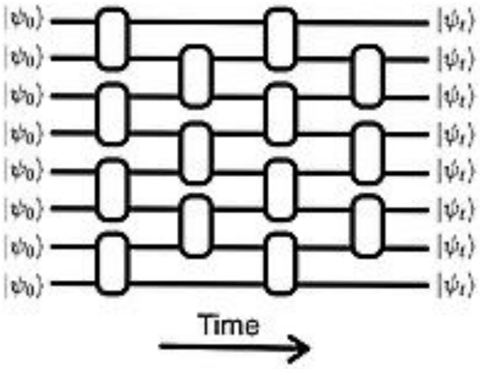
Performed operations

$$= O(\mathrm{poly}(\varepsilon))$$

Where blindness comes in?

Circuit

Input State



Resulting State = Circuit(State)

$$\text{state} \xrightarrow{\text{Circuit}} C\,(\text{state})$$

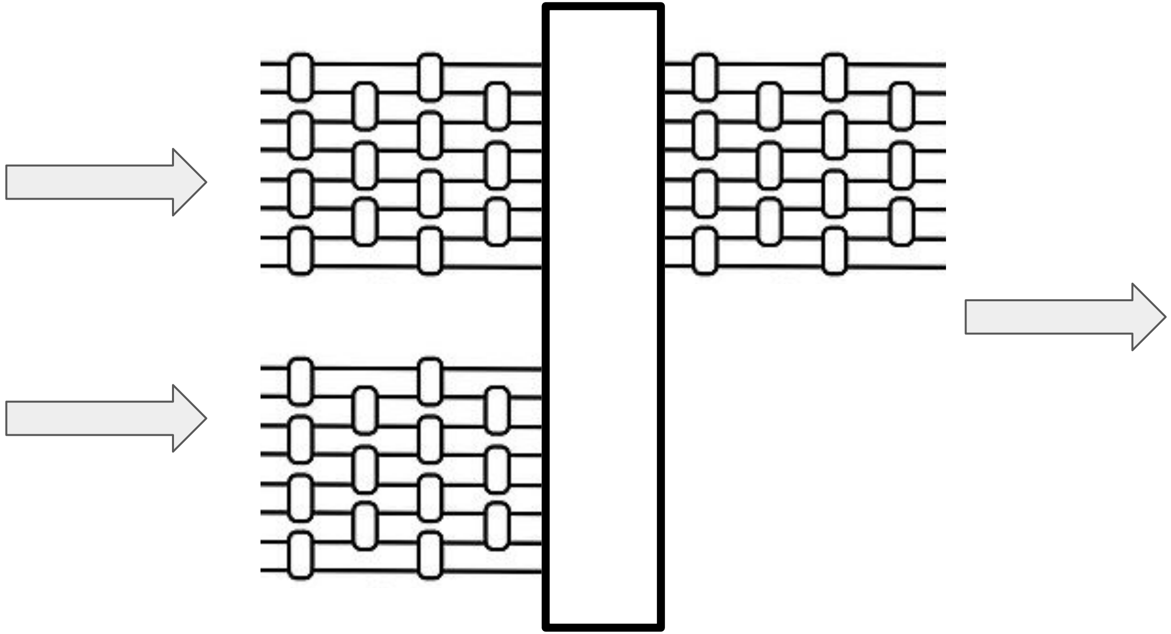
Where blindness comes in?

Universal circuit

*Universal form

Input State

Circuit



Resulting State

Circuit(State)

$$(\text{state, circuit}) \xrightarrow{\text{UC}} \text{circuit}(\text{state})$$

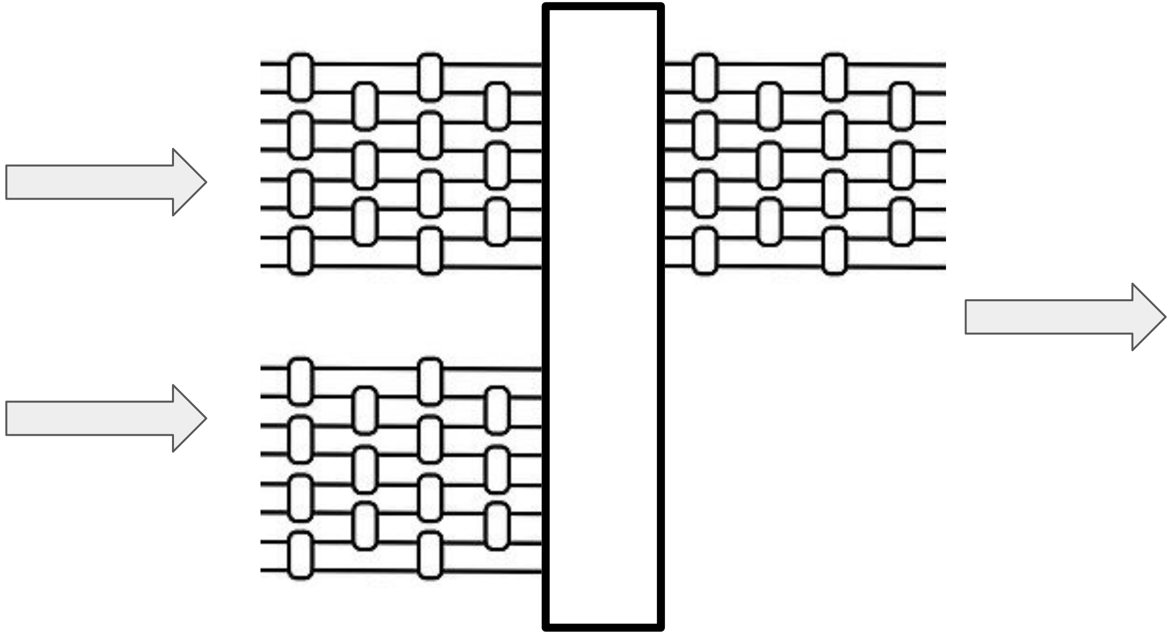
Where blindness comes in?

Universal circuit

*Universal form

Encrypted
Input State

Encrypted
Circuit



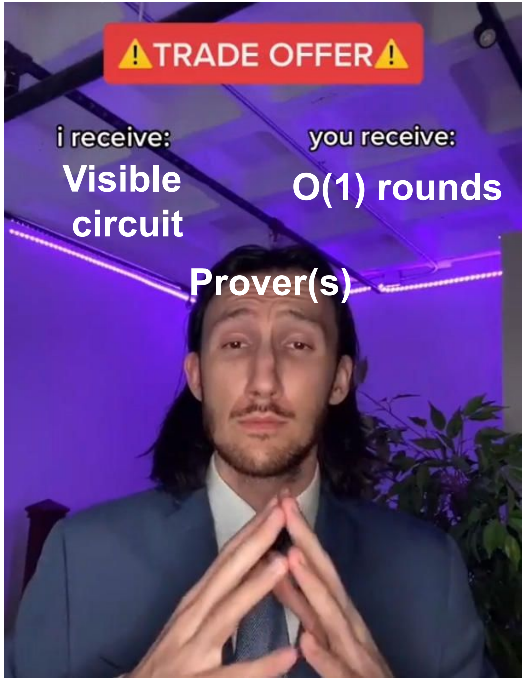
Encrypted
Resulting State
Circuit(State)

$$(\text{state, circuit}) \xrightarrow{\text{UC}} \text{circuit}(\text{state})$$

Dog-walker protocol

Blindness: is a property of delegation protocols, which informally states that the prover learns nothing* about the verifier's private circuit.

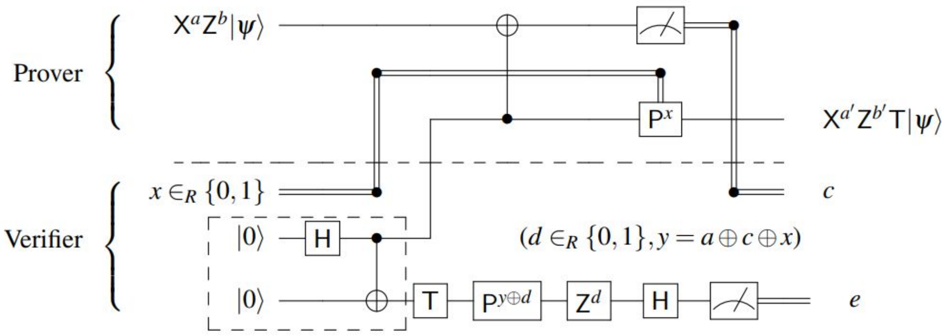
Prover is not blind to the computation in exchange for a reduction on the number of question rounds required to verify 😈/👼: O(1).



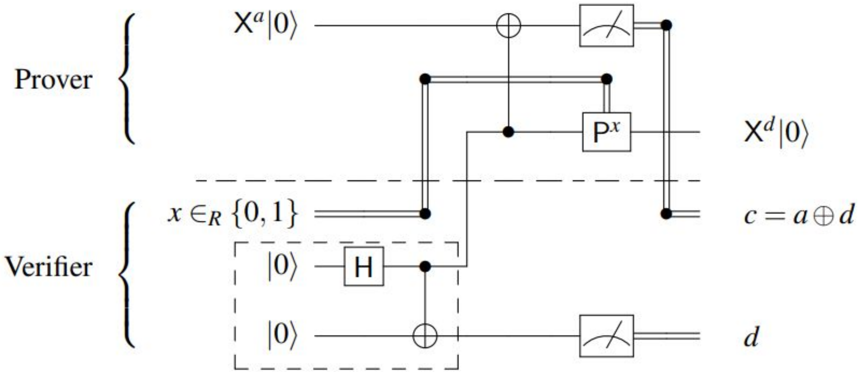
Conclusions

- Recap: Overview of non-local games and their relevance.
- Introduced a new framework: **Verifier on a Leash**, expanding the scope of verification protocols.
- Addressed both **trusted** and **non-trusted** PV scenarios.
  - Reduction of the **non-trusted PV** case to the trusted case.
- Presented the **scalability** of the protocol.
- Highlighted the **blindness property** of the new protocols.

T-gate Gadgets



- T gadget in a computational round
- Implements a T gate



- T gadget in a test round
- Implements the Identity

Broadbent, A. (2018). How to Verify a Quantum Computation. Theory OF Computing, 14(11), 1-37.