Overview of verified quantum computations using MBQC and recent progress

Harold Ollivier

QuantumTech@INRIA - QAT Team - https://qat.inria.fr

Outline

1. What is verification and why it is important
2. Short introduction to MBQC
3. Intuition and protocol construction for verified MBQC
4. Abstracting and generalizing
5. Lifting limitations and going toward practical solutions

01

What is verification and why it is important

An (overly simplified) example

Selling the Hadamard gate

Selling the Hadamard gate

First attempt

- $H\,|0\rangle = |+\rangle \quad H\,|1\rangle = |-\rangle$

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\left|0\right\rangle = \left|+\right\rangle \quad H\left|1\right\rangle = \left|-\right\rangle$
- $\Pr(\left|0\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$
- $\Pr(\left|1\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\left|0\right\rangle = \left|+\right\rangle \quad H\left|1\right\rangle = \left|-\right\rangle$
- $\Pr(\left|0\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$
- $\Pr(\left|1\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$

Second attempt

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\left|0\right\rangle = \left|+\right\rangle \quad H\left|1\right\rangle = \left|-\right\rangle$
- $\Pr(\left|0\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$
- $\Pr(\left|1\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$

Second attempt

- $H^2\left|\psi\right\rangle = \left|\psi\right\rangle$

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\left|0\right\rangle = \left|+\right\rangle \quad H\left|1\right\rangle = \left|-\right\rangle$
- $\Pr(\left|0\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$
- $\Pr(\left|1\right\rangle \mid H\left|0\right\rangle) = \frac{1}{2}$

Second attempt

- $H^2\left|\psi\right\rangle = \left|\psi\right\rangle$
- $\Pr\big(\left|0\right\rangle \mid H^2\left|0\right\rangle\big) = 1$
- $\Pr\big(\left|1\right\rangle \mid H^2\left|1\right\rangle\big) = 1$

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H|0\rangle = |+\rangle \quad H|1\rangle = |-\rangle$
- $\Pr(|0\rangle \,|H|0\rangle) = \frac{1}{2}$
- $\Pr(|1\rangle \,|H|0\rangle) = \frac{1}{2}$

Second attempt

- $H^2|\psi\rangle = |\psi\rangle$
- $\Pr\big(|0\rangle \,|H^2|0\rangle\big) = 1$
- $\Pr\big(|1\rangle \,|H^2|1\rangle\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\lvert 0\rangle = \lvert +\rangle \quad H\lvert 1\rangle = \lvert -\rangle$
- $\Pr(\lvert 0\rangle \mid H\lvert 0\rangle) = \frac{1}{2}$
- $\Pr(\lvert 1\rangle \mid H\lvert 0\rangle) = \frac{1}{2}$

Second attempt

- $H^2\lvert \psi\rangle = \lvert \psi\rangle$
- $\Pr\big(\lvert 0\rangle \mid H^2\lvert 0\rangle\big) = 1$
- $\Pr\big(\lvert 1\rangle \mid H^2\lvert 1\rangle\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

- Need to trust some model of the gate's behavior (e.g. fixed CPTP map)

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H|0\rangle = |+\rangle \quad H|1\rangle = |-\rangle$
- $\Pr(|0\rangle \,|\, H|0\rangle) = \frac{1}{2}$
- $\Pr(|1\rangle \,|\, H|0\rangle) = \frac{1}{2}$

Second attempt

- $H^2|\psi\rangle = |\psi\rangle$
- $\Pr\big(|0\rangle \,|\, H^2|0\rangle\big) = 1$
- $\Pr\big(|1\rangle \,|\, H^2|1\rangle\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

- Need to trust some model of the gate's behavior (e.g. fixed CPTP map)
- Esp. no ability to change behavior when used alone vs inside a computation

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H|0\rangle = |+\rangle \quad H|1\rangle = |-\rangle$
- $\Pr(|0\rangle \mid H|0\rangle) = \frac{1}{2}$
- $\Pr(|1\rangle \mid H|0\rangle) = \frac{1}{2}$

Second attempt

- $H^2|\psi\rangle = |\psi\rangle$
- $\Pr\big(|0\rangle \mid H^2|0\rangle\big) = 1$
- $\Pr\big(|1\rangle \mid H^2|1\rangle\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

- Need to trust some model of the gate's behavior (e.g. fixed CPTP map)
- Esp. no ability to change behavior when used alone vs inside a computation

Telling apart genuine and malicious implementations: online setup

Limitations to gate tomography:

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\left|0\right\rangle = \left|+\right\rangle \quad H\left|1\right\rangle = \left|-\right\rangle$
- $\Pr(\left|0\right\rangle \left| H \left|0\right\rangle\right.) = \frac{1}{2}$
- $\Pr(\left|1\right\rangle \left| H \left|0\right\rangle\right.) = \frac{1}{2}$

Second attempt

- $H^2\left|\psi\right\rangle = \left|\psi\right\rangle$
- $\Pr\big(\left|0\right\rangle \left| H^2 \left|0\right\rangle\right.\big) = 1$
- $\Pr\big(\left|1\right\rangle \left| H^2 \left|1\right\rangle\right.\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

- Need to trust some model of the gate's behavior (e.g. fixed CPTP map)
- Esp. no ability to change behavior when used alone vs inside a computation

Telling apart genuine and malicious implementations: online setup

Limitations to gate tomography:

- There is no guarantee that the behavior of gates will be repeatable

An (overly simplified) example

Selling the Hadamard gate

First attempt

- $H\,|0\rangle = |+\rangle \quad H\,|1\rangle = |-\rangle$
- $\Pr(|0\rangle\,|H\,|0\rangle) = \frac{1}{2}$
- $\Pr(|1\rangle\,|H\,|0\rangle) = \frac{1}{2}$

Second attempt

- $H^2\,|\psi\rangle = |\psi\rangle$
- $\Pr\big(|0\rangle\,|H^2\,|0\rangle\big) = 1$
- $\Pr\big(|1\rangle\,|H^2\,|1\rangle\big) = 1$

Telling apart genuine and malicious implementations: offline setup

Gate tomography

- Need to trust some model of the gate's behavior (e.g. fixed CPTP map)
- Esp. no ability to change behavior when used alone vs inside a computation

Telling apart genuine and malicious implementations: online setup

Limitations to gate tomography:

- There is no guarantee that the behavior of gates will be repeatable
- There is no guarantee that the behavior alone / inside a computation is the same (ie scalability pb)

Ideal Resource: Verified and Blind Quantum Computation

Verification (in Abstract Cryptography)

Ideal Resource: Verified and Blind Quantum Computation

- Allowed leakage
  - Upper bound on number of qubits n and depth

Verification (in Abstract Cryptography)

Ideal Resource: Verified and Blind Quantum Computation

- Allowed leakage
  - Upper bound on number of qubits n and depth
- Inputs
  - Client: Classical description of a computation C
  - Server: two bits b and c (with $b = c = 0$ as default)

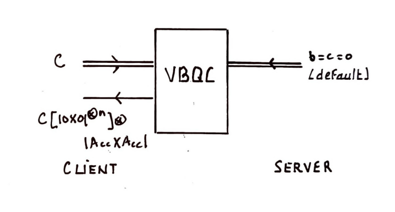
Verification (in Abstract Cryptography)

Ideal Resource: Verified and Blind Quantum Computation

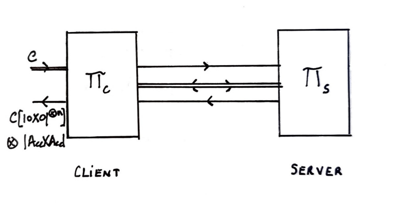
- Allowed leakage
  - > Upper bound on number of qubits n and depth
- Inputs
  - > Client: Classical description of a computation C
  - > Server: two bits b and c (with $b = c = 0$ as default)
- Computation
  - > if $b = 1$ it sends the allowed leakage to the Server, and if $c = 1$ is sent in return it sends $|\perp\rangle\langle\perp| \otimes |\mathrm{Rej}\rangle\langle\mathrm{Rej}|$ to the Client
  - > Otherwise it sends $C(|0\rangle\langle 0|^{\otimes n}) \otimes |\mathrm{Acc}\rangle\langle\mathrm{Acc}|$ to the Client

Ideal Resource: Verified and Blind Quantum Computation

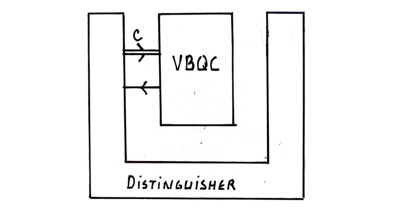
- Allowed leakage
  - > Upper bound on number of qubits n and depth
- Inputs
  - > Client: Classical description of a computation C
  - > Server: two bits b and c (with $b = c = 0$ as default)
- Computation
  - > if $b = 1$ it sends the allowed leakage to the Server, and if $c = 1$ is sent in return it sends $|\perp\rangle\langle\perp| \otimes |\mathrm{Rej}\rangle\langle\mathrm{Rej}|$ to the Client
  - > Otherwise it sends $C(|0\rangle\langle 0|^{\otimes n}) \otimes |\mathrm{Acc}\rangle\langle\mathrm{Acc}|$ to the Client

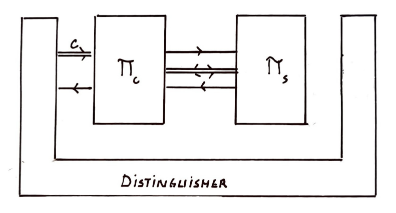


Protocol: Verified and Blind Quantum Computation

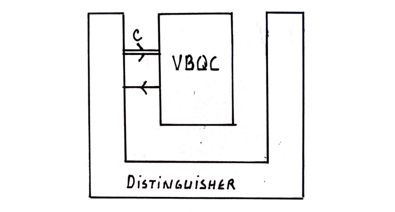


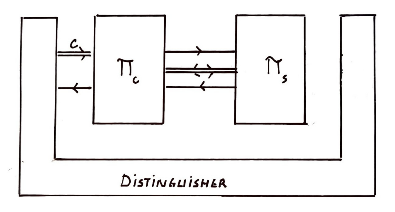
Correctness



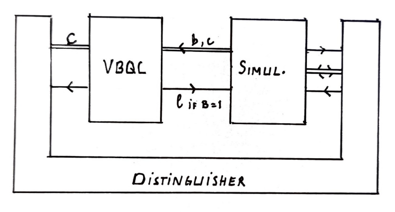


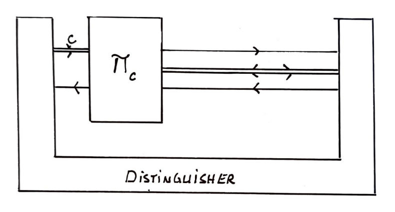
Correctness





Security

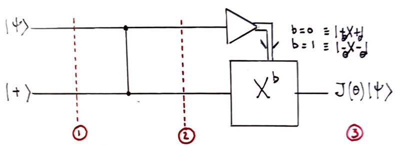




02

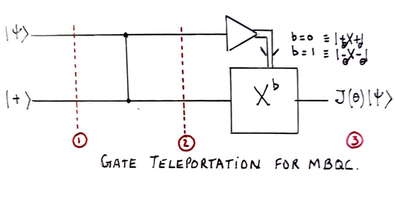
Short introduction to MBQC

Gate Teleportation



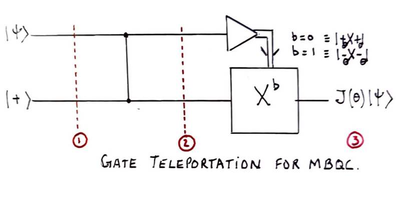
GATE TELEPORTATION FOR MBQC.

Gate Teleportation



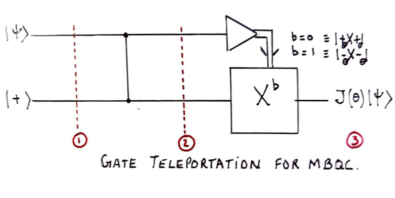
1. $(\alpha |0\rangle + \beta |1\rangle) \otimes |+_\theta\rangle$

Gate Teleportation



1. $(\alpha|0\rangle + \beta|1\rangle) \otimes |+_\theta\rangle$
2. $|+_\theta\rangle \otimes (\alpha|+\rangle + e^{i\theta}\beta|-\rangle)/\sqrt{2} + |-_\theta\rangle \otimes (-\alpha|+\rangle + e^{i\theta}\beta|-\rangle)/\sqrt{2}$

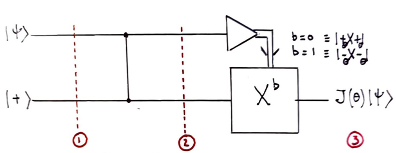
Gate Teleportation



1. $(\alpha|0\rangle + \beta|1\rangle) \otimes |+_\theta\rangle$
2. $|+_\theta\rangle \otimes (\alpha|+\rangle + e^{i\theta}\beta|-\rangle)/\sqrt{2} + |-_\theta\rangle \otimes (-\alpha|+\rangle + e^{i\theta}\beta|-\rangle)/\sqrt{2}$
3. $\alpha|+\rangle + \beta e^{i\theta}|-\rangle = HZ(\theta)(|0\rangle + \beta|1\rangle)$

Gate teleportation and universal computation

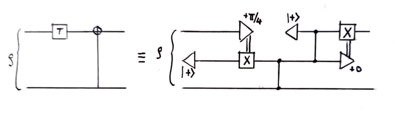
Gate Teleportation



GATE TELEPORTATION FOR MBQC.

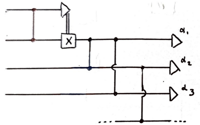
1. $(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes |+_\theta\rangle$
2. $|+_\theta\rangle \otimes (\alpha\,|+\rangle + e^{i\theta}\beta\,|-\rangle)/\sqrt{2} + |-_\theta\rangle \otimes (-\alpha\,|+\rangle + e^{i\theta}\beta\,|-\rangle)/\sqrt{2}$
3. $\alpha\,|+\rangle + \beta e^{i\theta}\,|-\rangle = HZ(\theta)(|0\rangle + \beta\,|1\rangle)$

Universality



UNIVERSALITY OF $J(\alpha)$ + C-Z

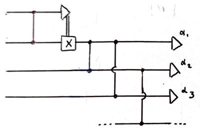
Pushing corrections to the end



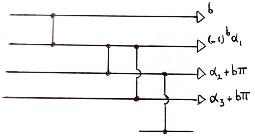
TYPICAL RESULT OF GT COMPILATION

MBQC as lazy implementation of GT

Pushing corrections to the end



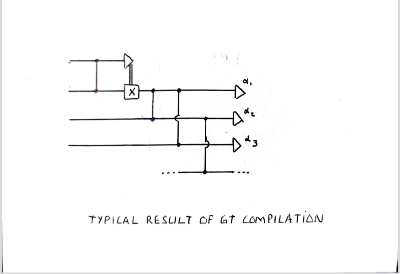
TYPICAL RESULT OF GT COMPILATION

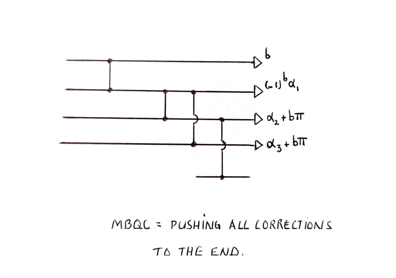


MBQC = PUSHING ALL CORRECTIONS TO THE END.

MBQC as lazy implementation of GT

Pushing corrections to the end





Summary

- Graph and partial order over vertices
- Flow
- Measurement angles (for the all-0 branch)

03

Intuition and protocol construction for verified MBQC

Intuition building

Preventing the Server to be malicious

Efficiency is not guaranteed

Preventing the Server to be malicious

Efficiency is not guaranteed

Making sure the Server is caught

Preventing the Server to be malicious

Efficiency is not guaranteed

Making sure the Server is caught

- Constantly test the behavior of the Server

Intuition building

Preventing the Server to be malicious

Efficiency is not guaranteed

Making sure the Server is caught

- Constantly test the behavior of the Server
- Make sure tests and computation look the same

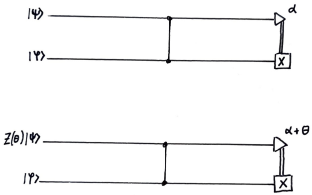
Blindness

Blindness

- MBQC works because Client and Server share a reference frame

Blindness

- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server

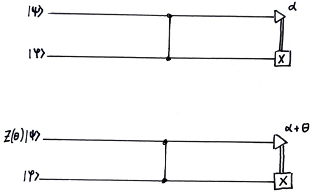


CONFUSING THE SERVER WITH HIDDEN REL. REFERENCE FRAME

Constructing a verification protocol with MBQC

Blindness

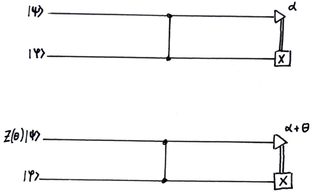
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations



CONFUSING THE SERVER WITH HIDDEN REL. REFERENCE FRAME

Blindness

- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations



CONFUSING THE SERVER WITH HIDDEN REL. REFERENCE FRAME

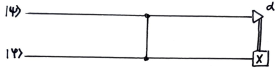
Trap insertion

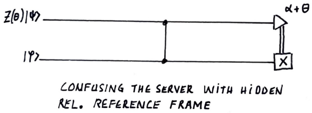
Blindness

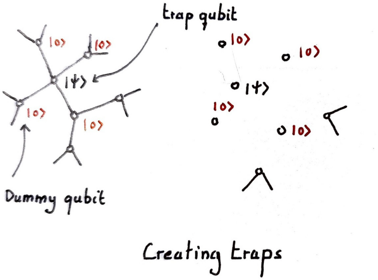
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

Trap insertion

- Creating traps





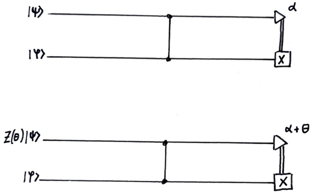


Blindness

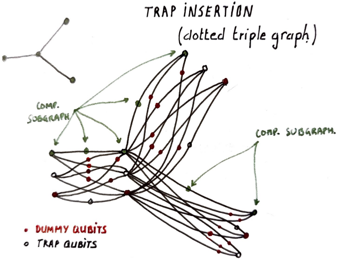
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

Trap insertion

- Creating traps
- Inserting traps



CONFUSING THE SERVER WITH HIDDEN REL. REFERENCE FRAME

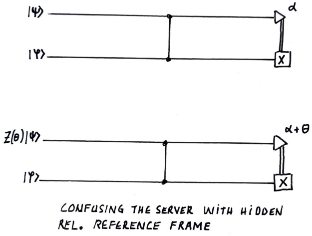


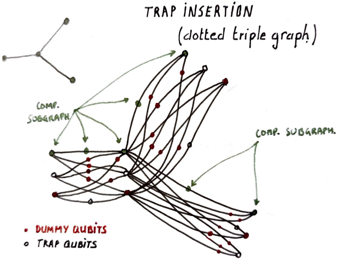
Blindness

- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

Trap insertion

- Creating traps
- Inserting traps
- Allows deviation detection with constant probability





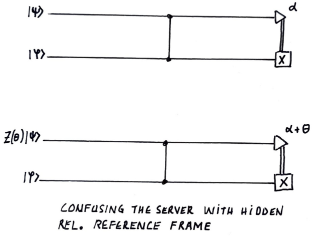
Blindness

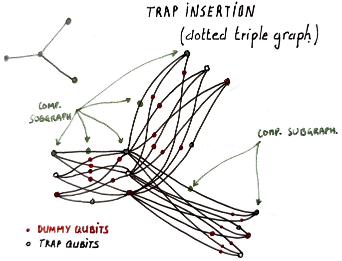
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

Trap insertion

- Creating traps
- Inserting traps
- Allows deviation detection with constant probability

Amplification





Blindness

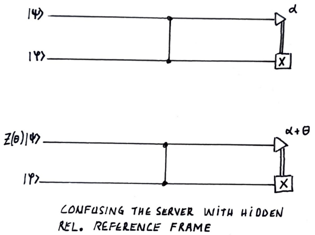
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

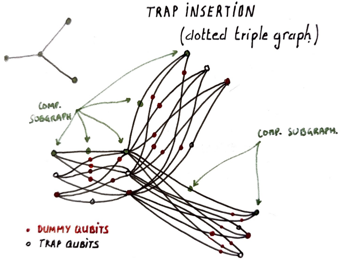
Trap insertion

- Creating traps
- Inserting traps
- Allows deviation detection with constant probability

Amplification

- Making sure that harmfull deviations are detected





Blindness

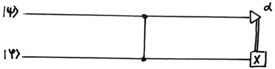
- MBQC works because Client and Server share a reference frame
- By sending $|+\rangle_\theta$ the client defines a relative RF unknown to the Server
- Blindness reduces attacks to convex combinations of Pauli deviations

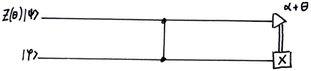
Trap insertion

- Creating traps
- Inserting traps
- Allows deviation detection with constant probability

Amplification

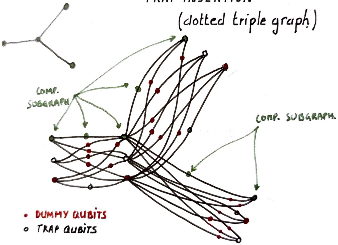
- Making sure that harmfull deviations are detected
- Using fault-tolerant encoding before trap insertion





CONFUSING THE SERVER WITH HIDDEN REL. REFERENCE FRAME

TRAP INSERTION (dotted triple graph)



• DUMMY QUBITS
∘ TRAP QUBITS

Limitations

Overhead

- Fault-tolerant encoding for amplification is costly
- Security competes with computing power (ie. for the number of live-qubits)

Limitations

Overhead

- Fault-tolerant encoding for amplification is costly
- Security competes with computing power (ie. for the number of live-qubits)

Robustness

- The fault-tolerant encoding does not protect from errors
- As soon as a single trap fails, the computation is aborted

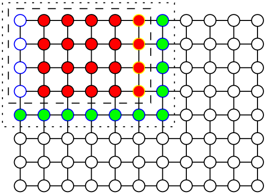
04

Abstracting and generalizing

Trappified canvas

Partial pattern

- G_P subgraph of G
- Input and output sets of nodes, I and O
- flow on G_P
- measurement angles ϕ_v

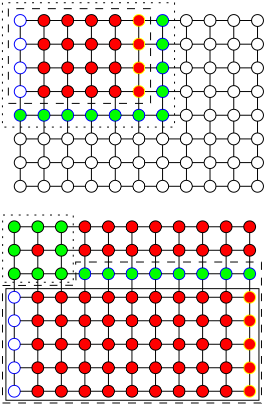


Partial pattern

- G_P subgraph of G
- Input and output sets of nodes, I and O
- flow on G_P
- measurement angles ϕ_v

Trappified canvas

- T partial pattern
- σ single-qubit product state on I
- $\mathcal{T}$ an efficiently computable probability distribution for X measurements of qubits it O
- τ a decision algorithms that takes a sample from $\mathcal{T}$ and outputs Pass or Fail



Trappified canvas

Partial pattern

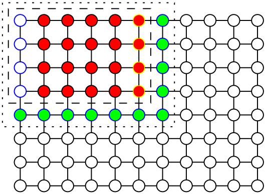
- G_P subgraph of G
- Input and output sets of nodes, I and O
- flow on G_P
- measurement angles ϕ_v

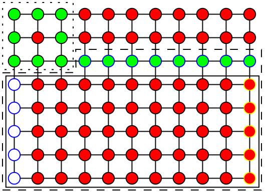
Trappified canvas

- T partial pattern
- σ single-qubit product state on I
- $\mathcal{T}$ an efficiently computable probability distribution for X measurements of qubits it O
- τ a decision algorithms that takes a sample from $\mathcal{T}$ and outputs Pass or Fail

Trappified scheme

- A collection of canvas and an embedding algorithm that maps computations to patterns given a trappified canvas





Conditions for verification

Pauli detection

P ϵ-detects $\mathcal{E} \subset \mathcal{G}_V$ if

$$\forall E \in \mathcal{E}, \sum_{T \in P} \Pr[\tau(t) = 1, T] \geq 1 - \epsilon$$

The probability is over the choice of canvas in the scheme and samples of the trap measurements t

Pauli insensitivity

P is δ-insensitive to $\mathcal{E} \subset \mathcal{G}_V$ if

$$\forall E \in \mathcal{E}, \sum_{T \in P} \Pr[\tau(t) = 0, T] \geq 1 - \delta$$

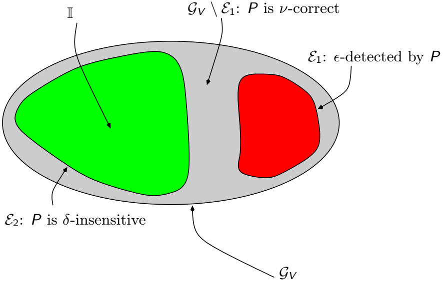
Pauli correctness

P is ν-correct on $\mathcal{E} \subset \mathcal{G}_V$ if,

$$\forall E \in \mathcal{E}, \ \forall C, \ \forall T \in P, \max_{\psi} \|(\tilde{C}_{T,E} - C) \otimes \mathbb{I}_R \, |\psi\rangle\langle\psi| \, \|_{\mathrm{tr}} \leq \nu$$

C is the intended computation, $\tilde{C}_{T,E}$ is the pattern followed by the deviation E

Conditions for verification



Detection implies verifiability

$\mathcal{E}_1$, $\mathcal{E}_2$ two pauli deviations sets with $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$ and $\mathbb{I} \in \mathcal{E}_2$. If P trappified scheme

- ϵ-detects $\mathcal{E}_1$,
- δ-insensitive to $\mathcal{E}_2$,
- ν-correct on $\mathcal{G}_V \setminus \mathcal{E}_1$ P allows for $\delta + \nu$ correct and $\max(\epsilon, \nu)$ secure deletgate quantum computing in AC.

05

Lifting limitations and going toward practical solutions

What is it good for?

Separate concerns

- Trap design is easier and decoupled from the security proof
- Amplification process can be changed

Impact

- Traps based on any measurement of stabilizer generator of the graph work
- Allows to diversify the trappified canvas and adapt them to specific setups
  - Robust verification
  - Multi-party computation
  - Rotation-only clients
  - Fault-tolerant delegation of quantum computation

A simple robust verification protocol

Changing only the amplification procedure

A simple robust verification protocol

Changing only the amplification procedure

- Traps and Computations are in different rounds (test and computation rounds)

A simple robust verification protocol

Changing only the amplification procedure

- Traps and Computations are in different rounds (test and computation rounds)
- Amplification is done through majority voting (works only for BQP computations)

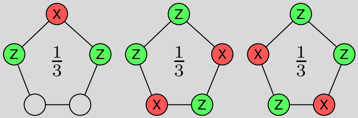
A simple robust verification protocol

Changing only the amplification procedure

- Traps and Computations are in different rounds (test and computation rounds)
- Amplification is done through majority voting (works only for BQP computations)

Protocol

- Test rounds correspond to graph coloring



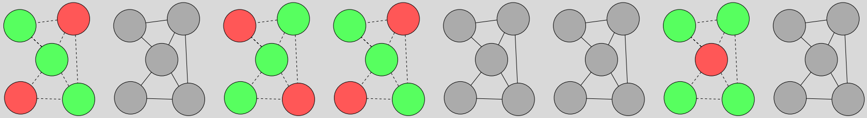
A simple robust verification protocol

Changing only the amplification procedure

- Traps and Computations are in different rounds (test and computation rounds)
- Amplification is done through majority voting (works only for BQP computations)

Protocol

- Test rounds correspond to graph coloring
- Interleave computation and test rounds at random



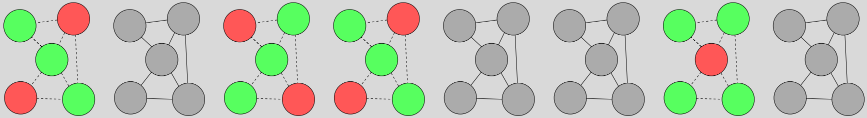
A simple robust verification protocol

Changing only the amplification procedure

- Traps and Computations are in different rounds (test and computation rounds)
- Amplification is done through majority voting (works only for BQP computations)

Protocol

- Test rounds correspond to graph coloring
- Interleave computation and test rounds at random



Benefit

- Robust up to 25% failure of test rounds
- Still not scalable (cf. Fault-tolerant version)

Conclusion

1. Practical implementations are possible
2. Protocols scale
3. Open questions
   - Optimized schemes
   - Low overhead verification for sampling
   - Lowering the communication complexity
   - Time to insert verification into HW roadmaps

06

Thanks you! (questions?)

References

- MBQC:
  - The one-way quantum computer - a non-network model of quantum computation, R Raussendorf, D Browne, H Briegel, arXiv:quant-ph/0108118
  - One-way Quantum Computation - a tutorial introduction, D Browne, H Briegel, arXiv:quant-ph/0603226
  - The Measurement Calculus, E Kashefi, V Danos, P Panangaden, arXiv:quant-ph/0412135
- Abstract cryptography:
  - Abstract Cryptography, U Maurer, R Renner, `https://crypto.ethz.ch/publications/files/MauRen11.pdf`
  - Cryptographic security of quantum key distribution, C Portmann, R Renner, arXiv:1409.3525
- Background on verification
  - Interactive Proofs for Quantum Computations, D Aharonov, M Ben-Or, E Eban, U Mahadev, arXiv:1704.04487
  - Verification of Quantum Computation: An Overview of Existing Approaches, A Gheorghiu, T Kapourniotis, E Kashefi

References

- Definition of VBQC ideal resource:
  - Composable security of delegated quantum computation, V Dunjko, J Fitzsimons, C Portmann, and R Renner, arXiv:1301.3662
- VBQC protocols:
  - Unconditionally Verifiable Blind Quantum Computation, J Fitzsimons and E Kashefi, arXiv:1203.5217
  - Optimised resource construction for verifiable quantum computation, E Kashefi and P Wallden, arXiv:1510.07408
- Recent work:
  - Unifying Quantum Verification and Error-Detection: Theory and Tools for Optimisations, T Kapourniotis, E Kashefi, D Leichtle, L Music, HO, arXiv:2206.00631
  - Verifying BQP Computations on Noisy Devices with Minimal Overhead, D Leichtle, L Music, E Kashefi, HO, arXiv:2109.04042
  - Asymmetric Quantum Secure Multi-Party Computation With Weak Clients Against Dishonest Majority, T Kapourniotis, E Kashefi, D Leichtle, L Music, HO, arXiv:2303.08865