Interactive Proofs For Quantum Computation

Elad Eban
The Hebrew University of Jerusalem

Joint work with Dorit Aharonov and Michael Ben-Or

Motivating questions

Assuming

BQP≠BPP

- **D-Wave is trying to sell me a 100 qubit quantum computer !**
- How can I verify it’s indeed a quantum computer?
- Is it possible to **delegate** computations to an ***untrusted*** company?
- How can an experimentalist check that his system is a quantum computer?

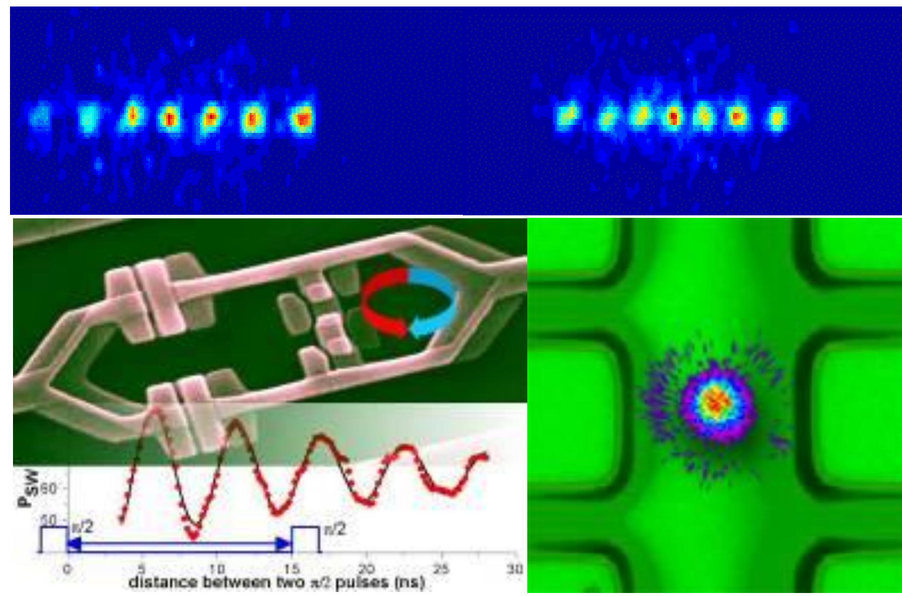
The Quantum Computing Company™

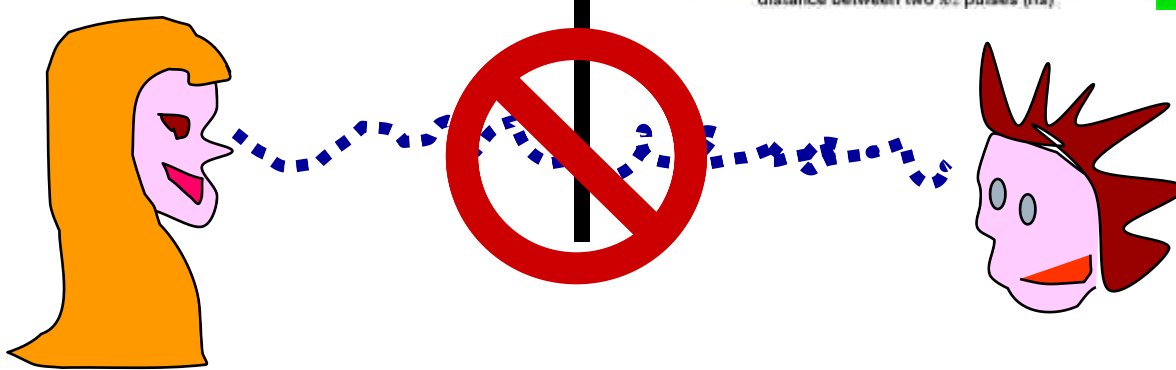
Quantum physics is different!
Seems impossible to test

Theory

Experiment

- To compute predictions we need a quantum computer!
- **Is QM (out of BPP) falsifiable? [Vazirani'07]**
- Inability to test QM most interesting experiments



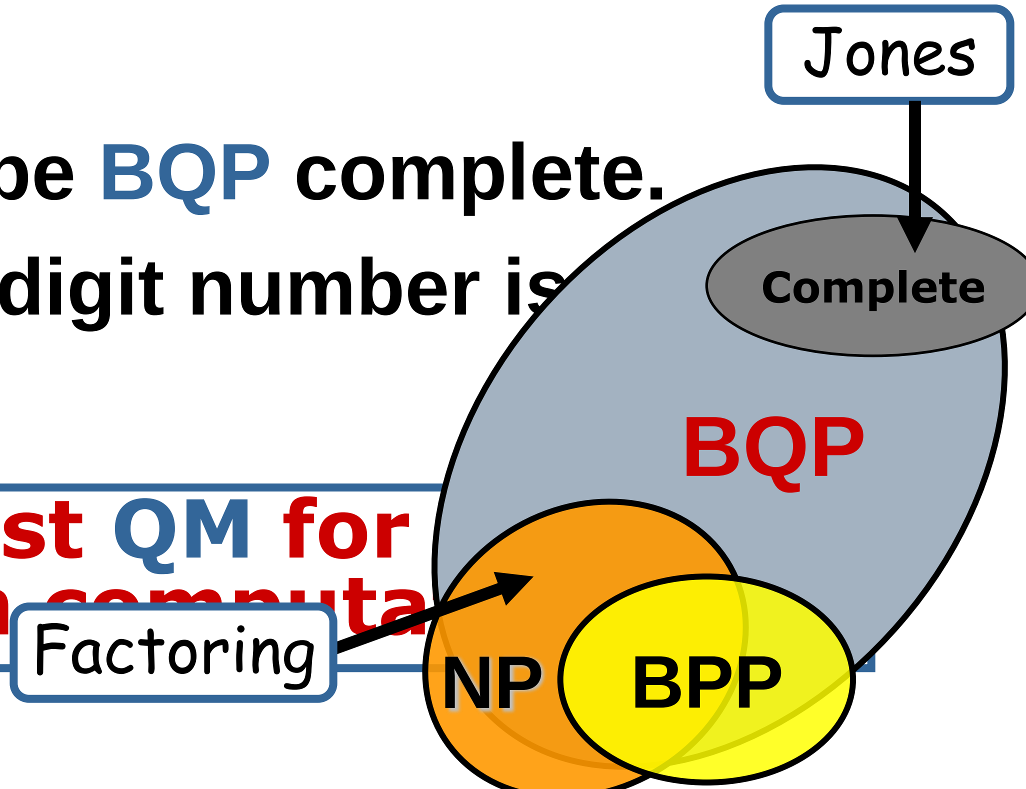


Shor's factoring Algorithm

- Shor's factorization provides a partial answer to our questions.

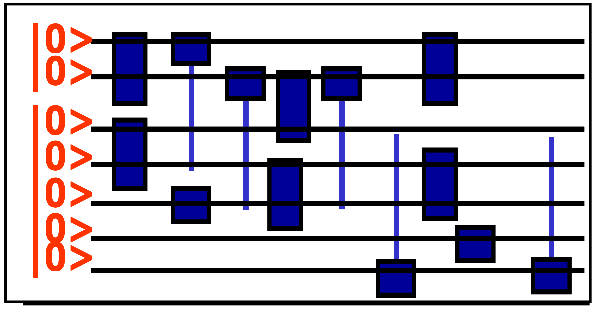
- **But:**
- **Not believed to be BQP complete.**
- **Factoring a 100 digit number is**

We want to test QM for
quantum computa



Q-CIRCUIT: A **BQP** complete problem

- **Q-CIRCUIT:**
  Input: a quantum circuit given by gates: $U = U_T \ldots U_1$, acting on **n** qubits.



Output: distinguish between:

$$\text{Q-CIRCUIT}_{\text{YES}} : Pr(\text{output} = 1) \geq \frac{2}{3}$$

$$\text{Q-CIRCUIT}_{\text{NO}} : Pr(\text{output} = 0) \geq \frac{2}{3}$$

Our goal:

Design a mechanism to test the correctness of the output of a polynomial **Quantum circuit**.

The problem:

How can we test the final answer, **without simulating** **BQP** efficiently?

New approach:

Treat interaction with quantum computers as

Interactive Proofs

Shor's Algorithm as an Interactive proof [GMR85]

Verifier

Q: Convince me that the 3^{rd} digit of the largest factor of **N** is **0** ?

A: (**$x<y$**) s.t. **$N=xy$**

Prover

BPP restricted

BQP power suffices

New model:
Quantum Prover Interactive Proof (**QPIP**)

Verifier

$|101\rangle$

$|111\rangle$

Prover

BPP* restricted

+ **O(1)** qubits

BQP restricted

Our results

- **Theorem (simplified)**:
  Any **BQP** language has a **QPIP** protocol.
  (We show one for the complete problem **Q-CIRCUIT**)
- **Main Result: Fault Tolerant QPIP**
  The above holds in the presence of noise.

- ***"Blind"*** **QPIP:** the same but the prover gains no information about neither the function being computed nor the input.

Simple **QPIP** for **Q-CIRCUIT**

Main idea: Authentication

- We would like to force the prover to **correctly perform a given computation**.
- Let us first try something easier: **force the prover to do nothing .**
- How do we check that an unknown state is unaltered?
- This is exactly what **Quantum Authentication Scheme (QAS)** does**.**

me

Encoding

$$U_k|\varphi\rangle|000\rangle$$

- Alice war .
- Alice enc t key
- Eve migh ed ~~message~~ **state**
- **Security** ability any

QAS solves the case of the identity circuit!

Prover manages to convince verifier he did nothing

Verifier

Alice

|m>

|m’>

Bob

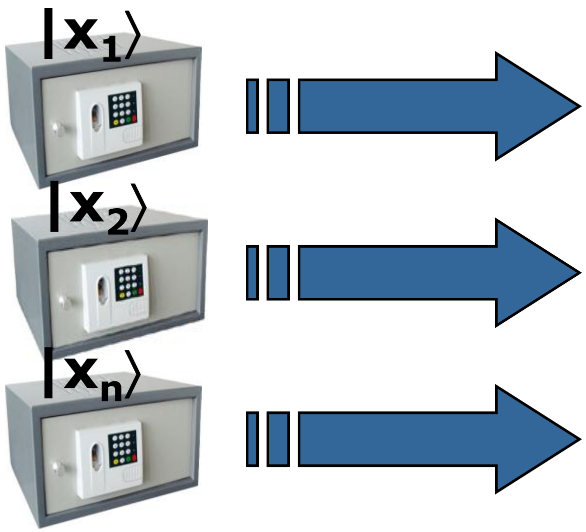
Prover

Simple QPIP for Q-CIRCUIT

Use the **prover as an untrusted storage device.**

Initialization

Verifier



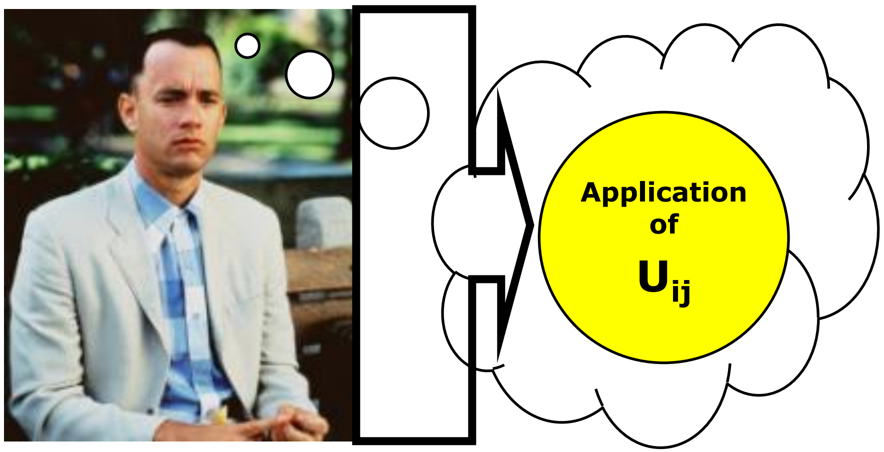
Prover

Simple QPIP for Q-CIRCUIT

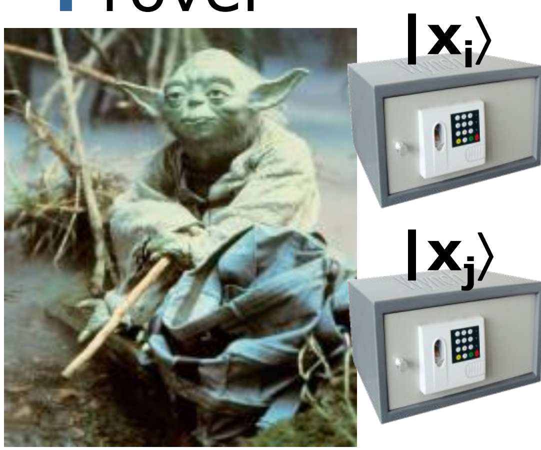
Use the **prover as an untrusted storage device.**

Performing operations

Verifier



Prover



Checks that the state are correctly authenticated

Simple **QPIP** for **Q-CIRCUIT**

Use the **prover as an untrusted storage device.**

Retrieving outcome

Verifier

Prover

$|\mathbf{x_1}\rangle$

Making a fault tolerant **QPIP**

Fault tolerance

- The **verifier cannot apply error correction** since error correction must be applied in **parallel** and verifier can only work **sequentially**.

- The **prover cannot apply the computation**, since he does not know the code!

Fault tolerant QPIP: Ingredients for the solution

- **Our goal** is to enable the **prover** to apply gates on encoded states without knowing the code!
- **Then** (almost) standard fault tolerant techniques could be used.

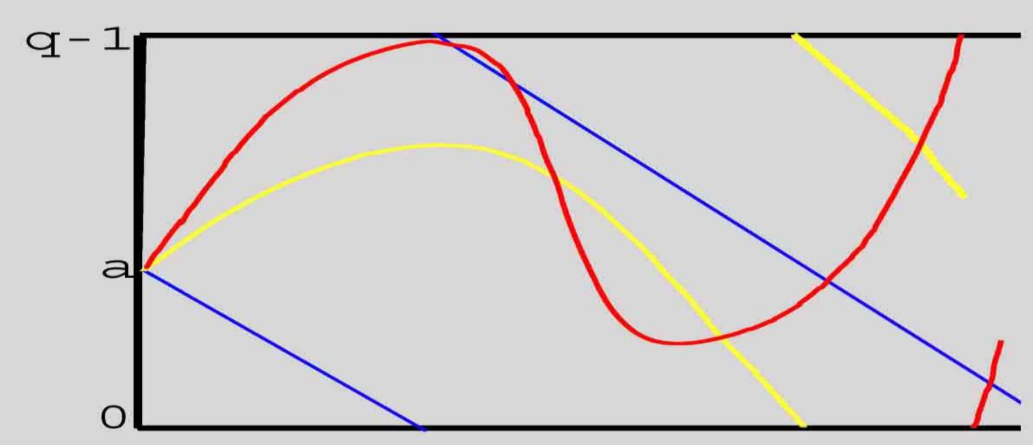
Recall similar questions were handled:

  - In [BGW] Multiparty computation, by using **Reed-Solomon codes**.
  - For **quantum** multiparty computation, [CGS] used **quantum RS codes**
  - Improved by [BCGHS06] using **randomized quantum RS codes**

- Constructions rely on the algebraic structure of the codes.

QAS based on randomized Quantum Reed-Solomon codes

[BCGHS06]



- The encoding of $a \in F_q$ is:

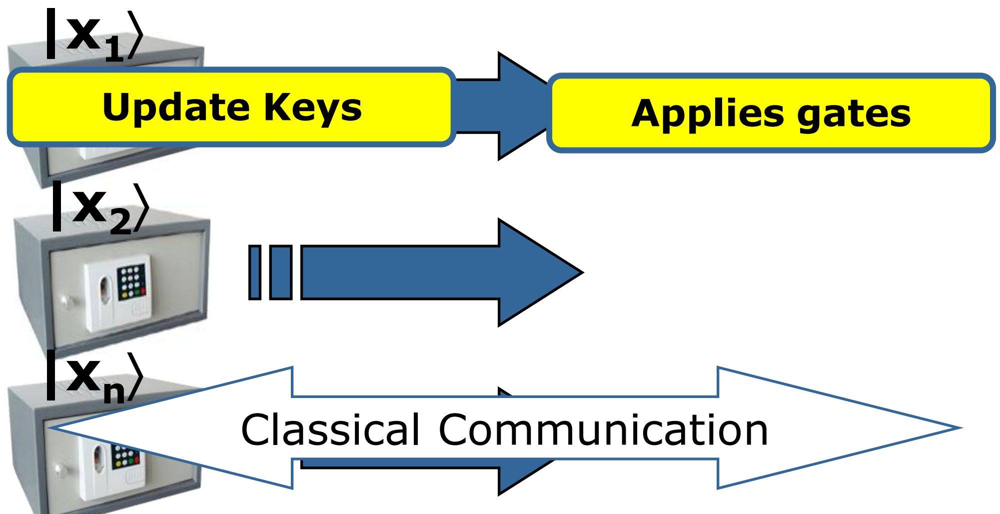
$$|s_a\rangle = \sum_{\substack{g:\deg(g)\le d \\ g(0)=a}} |g(\alpha_1), g(\alpha_2)\ldots, g(\alpha_m)\rangle$$

- For the **QAS:**
  - Apply a **random sign** (±1) to each register
  - Apply a **random Pauli's** to each register
  - The secret **KEY** is the sequence of Pauli's and signes
- **Main point:**
  - Prover can apply gates without knowing the key, **ignoring randomization**!
  - **Verifier modifies key** to compensate for the prover **ignorance**.
- A universal set of gates can be applied this way (communication with prover is needed)

Reed-Solomon based **QPIP**

Computation

Verifier



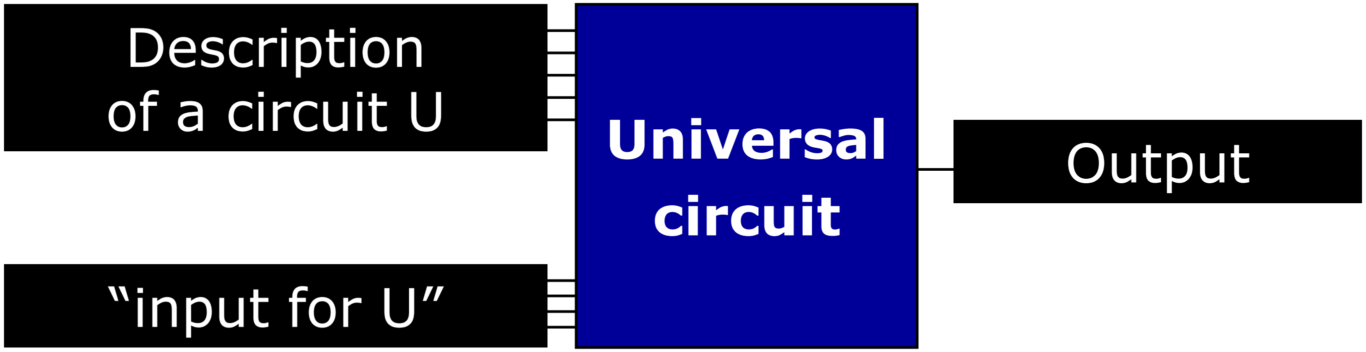
Prover

Information from the prover is checked to be correctly authenticated

Now that all the computation is done by the prover, standard fault tolerant techniques are applied

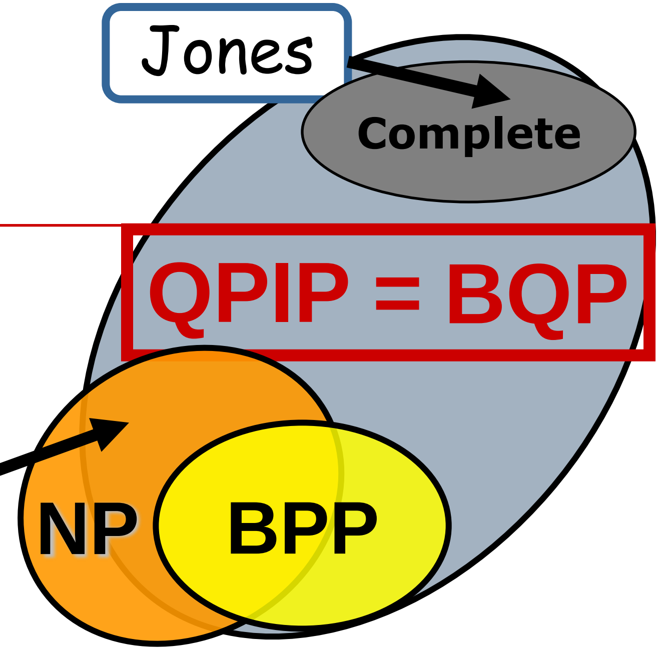
Making the QPIP blind

- **Task**:
  Hide **circuit** and **input** from the prover
- Both QPIP hide the **input** from the prover.
- **Solution**:
  Always apply a **universal quantum circuit**, plug the description of the circuit as part of the input.



Conclusions
&
Open Questions

To conclude:
The power of interaction



Shor's algorithm provides partial answers, but not enough.

We defined QPIP:
A model of **interaction between quantum** and almost **classical systems**.

Partial solution to Aaronson 25$-chalange

We provided a protocol for a **BQP complete** problem.
It is **fault tolerant**, **blind** and allows tackling all our motivating questions

Open Question

- **Major open question:** Do the same results hold with a completely Classical Verifier **QPIP**?

- A (possibly) easier question:
  Classical Verifier **two-prover** **QPIP**?
  - <u>Independently</u>: Broadbent, Fitzsimons & Kashefi [FOCS09] proved similar results. From very different motivations, using different techniques.
  - They also made some advance on the above question: They proved the result with are **entangled** prover.


The End
thank you
©Taken Jan 3rd, by Michal Feldman at -22c

Bibliography

- **[AB96] [AB99]** Fault tolerant Quantum computation with constant error.
- [**AS06**] P. Arrighi and L. Salvail. Blind Quantum Computation.
- [**BCG02**] H. Barnum, C. Cr´epeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of Quantum Messages.
- [**BFK08**] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation.
- [**BCGHS06**] M. Ben-Or, C. Cr´epeau, D. Gottesman, A. Hassidim, and A. Smith. Secure Multiparty Quantum Computation with (Only) a Strict Honest Majority.
- **[BGW]** Completeness theorems for non-cryptographic fault-tolerant distributed computation
- [**Chi01**] A.M. Childs. Secure assisted quantum computation.
- [**GMR85**] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems.
- [**Vaz07**] U. Vazirani. Computational constraints on scientific theories: insights from quantum computing. http://www.cs.caltech.edu/˜schulman/Workshops/CS-Lens-2/cs-lens-2.html, 2007.
- [**Sho97**] PW Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.