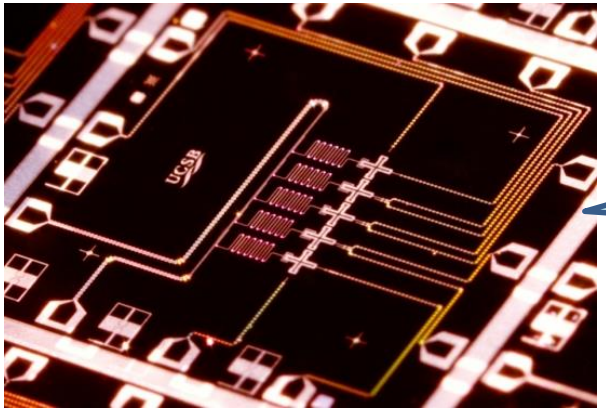
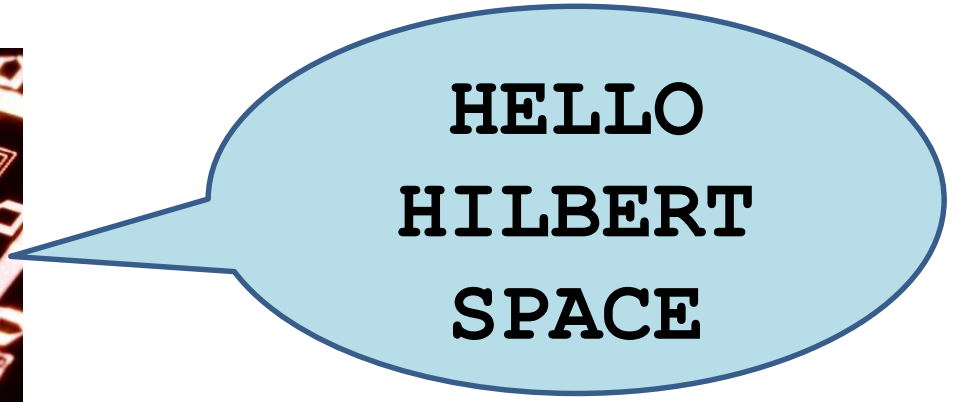
Quantum Supremacy and its Applications





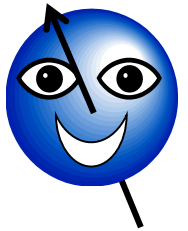
Scott Aaronson (University of Texas at Austin)

USC, October 11, 2018

Based on joint work with Lijie Chen (CCC'2017, arXiv:1612.05903) and on forthcoming work

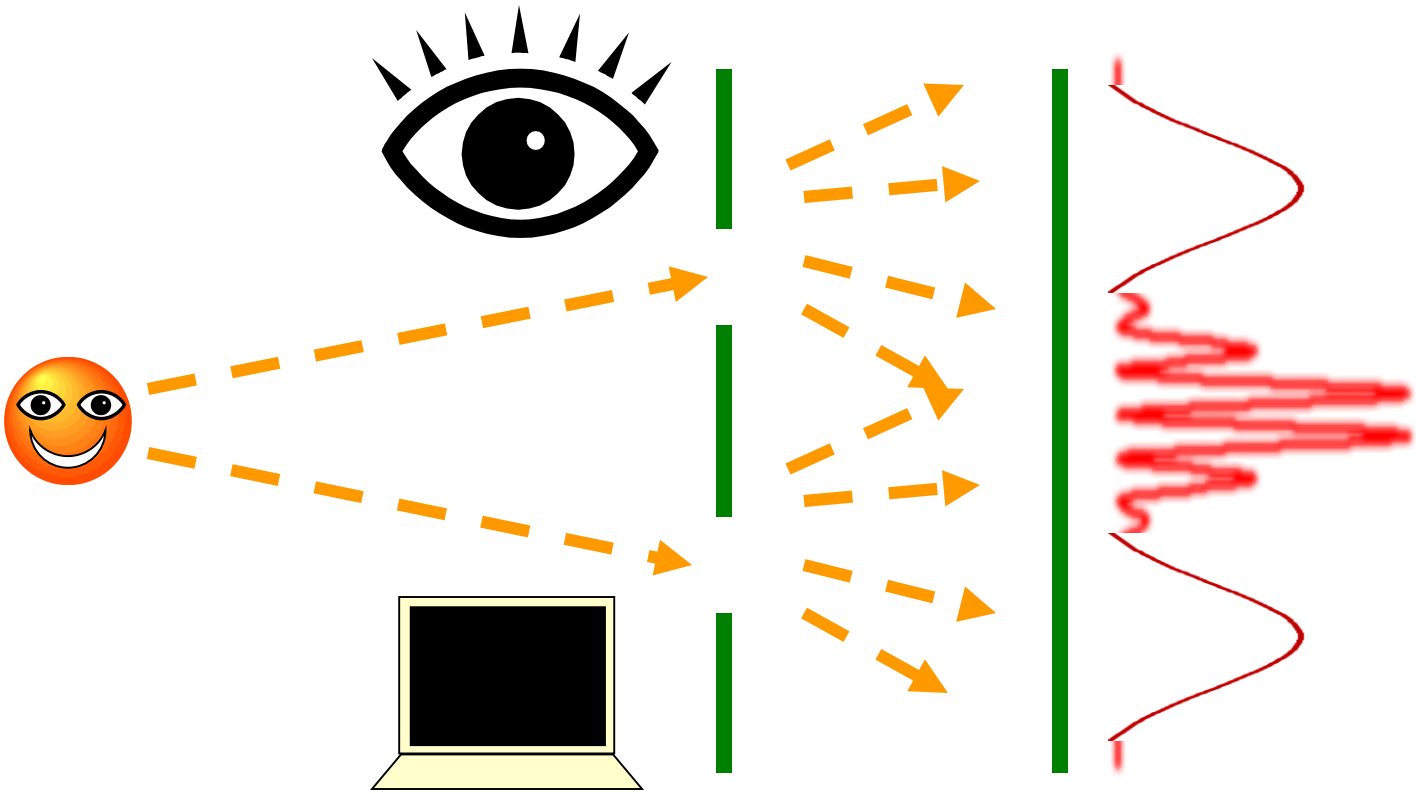
Papers and slides at www.scottaaronson.com

Quantum Mechanics



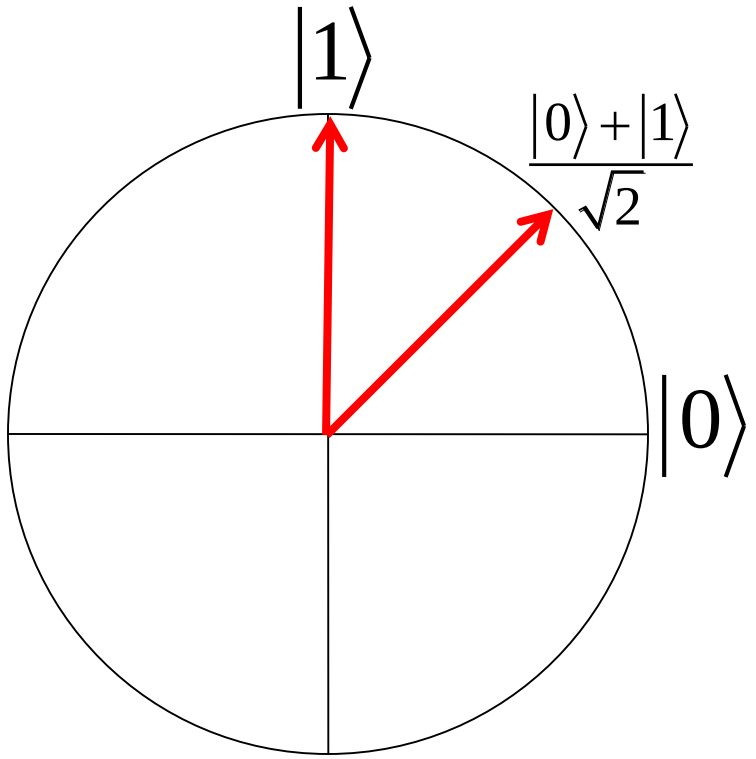
"Probability theory with minus signs"

(Nature seems to prefer it that way)



The key claim of quantum mechanics is that, if an object can be in two distinguishable states, call them $|0\rangle$ or $|1\rangle$, then it can also be in a **superposition**

$$\mathbf{a|0\rangle + b|1\rangle}$$

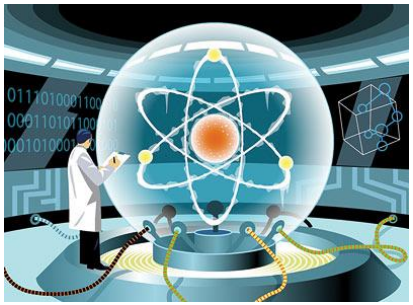


Here a and b are complex numbers called **amplitudes** satisfying $|a|^2+|b|^2=1$

If we observe, we see
- $|0\rangle$ with probability $|a|^2$
- $|1\rangle$ with probability $|b|^2$

Also, the object **collapses** to whichever outcome we see

Quantum Computing



A general **entangled** state of n qubits requires ~2^n amplitudes to specify:

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$$

Presents an obvious practical problem when using conventional computers to **simulate** quantum mechanics

Interesting

Feynman 1981: So then why not tur… build computers that **themselves** exploit superposition

Shor 1994: Such a computer could do more than simul… QM—e.g., it could factor integers in polynomial time

QUANTUM SUPREMACY

#1 Application of quantum computing: Disprove QC skeptics! (And the Extended Church-Turing Thesis)

Might actually be able to achieve in the next couple years, e.g. with Google's 72-qubit Bristlecone chip. "Obviously useless for anything else," but who cares?

The Sampling Approach

Put forward by Terhal-Divincenzo 2004 (constant-depth quantum circuits), A.-Arkhipov 2011 (BosonSampling), Bremner-Jozsa-Shepherd 2011 (IQP Model), and others

Consider problems where the goal is to **sample** from a desired distribution over n-bit strings

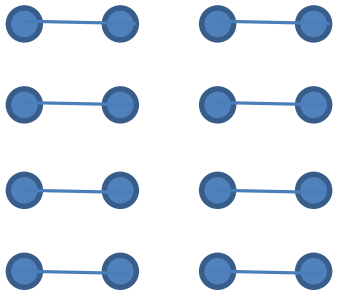
Compared to problems with a single valid output (like FACTORING), sampling problems can be

(1) Easier to solve with near-future quantum devices, and

(2) Easier to argue are hard for classical computers!

(We "merely" give up on: obvious applications, a fast classical way to verify the result…?)

The Random Quantum Circuit Proposal



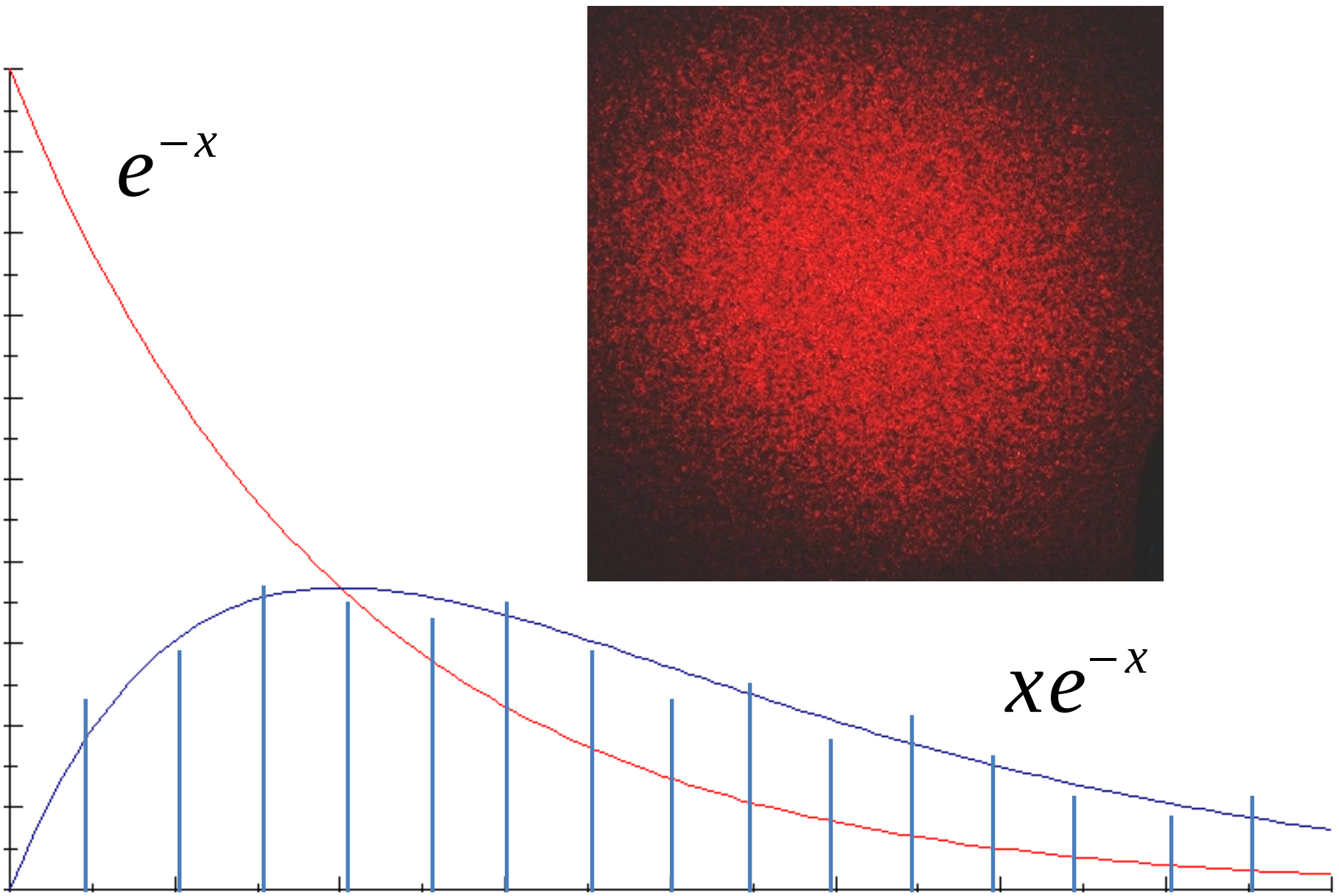
Generate a quantum circuit C on n qubits in a $\sqrt{n} \times \sqrt{n}$ lattice, with d layers of random nearest-neighbor gates

Apply C to $|0\rangle^{\otimes n}$ and measure. Repeat T times, to obtain samples $x_1,\ldots,x_T$ from $\{0,1\}^n$

Check whether $x_1,\ldots,x_T$ solve the "Heavy Output Generation" (HOG) problem—e.g., do at least 2/3 of the x_i's have more than the median probability?

(takes classical exponential time, which is OK for n≈70)

Publish C. Challenge skeptics to generate samples passing the test in a reasonable amount of time


e^{-x}
xe^{-x}

Our Strong Hardness Assumption

There's no polynomial-time classical algorithm A such that, given a uniformly-random quantum circuit C with n qubits and m>>n gates,

$$\Pr_C\left[A(C) \text{ guesses whether } \left|\langle 0|^{\otimes n} C |0\rangle^{\otimes n}\right|^2 > \text{median} \right] \geq \frac{1}{2} + \Omega\left(2^{-n}\right)$$

Note: There *is* a polynomial-time classical algorithm that guesses with probability $\approx \frac{1}{2} + \frac{1}{4^m}$

(just expand $\langle 0|^{\otimes n} C |0\rangle^{\otimes n}$ out as a sum of 4^m terms, then sample a few random ones)

Theorem: Assume SHA. Then given as input a random quantum circuit C, with n qubits and m>>n gates, there's no polynomial-time classical algorithm that even **passes our statistical test for C-sampling** w.h.p.

Proof Sketch: Given a circuit C, first "hide" which amplitude we care about by applying a random XOR-mask to the outputs, producing a C' such that $\langle 0|^{\otimes n} C'|z\rangle = \langle 0|^{\otimes n} C|0\rangle^{\otimes n}$

Now let A be a poly-time classical algorithm that passes the test for C' with probability ≥ 0.99. Suppose A outputs samples $x_1,\ldots,x_T$. Then if $x_i = z$ for some $i \in [T]$, guess that

$$\left|\langle 0|^{\otimes n} C|0\rangle^{\otimes n}\right|^2 \geq \text{median}$$

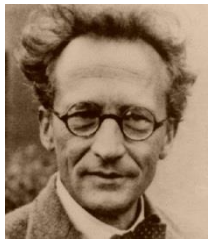
Otherwise, guess that with probability $\frac{1}{2} - \frac{T}{2^{n+1}}$

Violates SHA!

Time-Space Tradeoffs for Simulating Quantum Circuits

Given a general quantum circuit with n qubits and m>>n two-qubit gates, how should we simulate it classically?

"Schrödinger way":

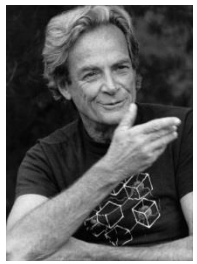


Store whole wavefunction

$O(2^n)$ memory, $O(m2^n)$ time

n=40, m=1000: Feasible but requires TB of RAM

"Feynman way":



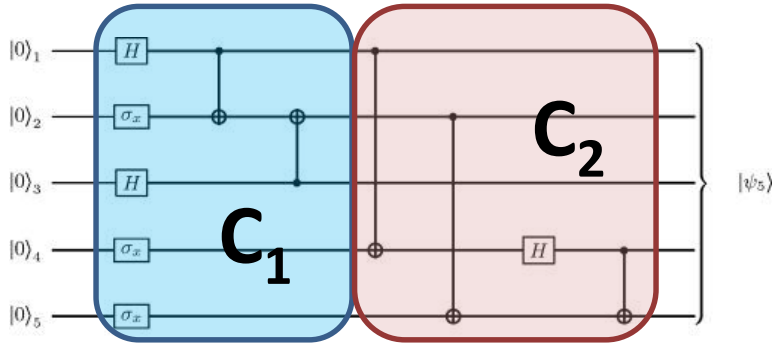
Sum over paths

$O(m+n)$ memory, $O(4^m)$ time

n=40, m=1000: Infeasible but requires little RAM

Best of both worlds?

Theorem: Let C be a quantum circuit with n qubits and d layers of gates. Then we can compute each transition amplitude, $\langle x|C|y\rangle$, in $d^{O(n)}$ time and O(nd) memory



Proof: Savitch's Theorem! Recursively divide C into two chunks, C_1 and C_2, with d/2 layers each. Then

$$\langle x|C|y\rangle = \sum_{z\in\{0,1\}^n} \langle x|C_1|z\rangle\langle z|C_2|y\rangle$$

This recursive approach is now being used in state-of-the-art classical simulations of quantum circuits (e.g. at IBM, Alibaba)

But it *still* doesn't falsify the SHA! Why not?

But if these sampling-based supremacy experiments work, they'll just produce mostly-random bits, which is *obviously* useless…

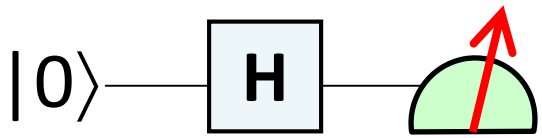
1101000011010011110110110011001100010100100100

Certified Random Bits: Who Needs 'Em?

For private use:
Cryptographic keys (a big one!)

For public use:
Election auditing, lotteries, parameters for cryptosystems, zero-knowledge protocols, proof-of-stake cryptocurrencies…

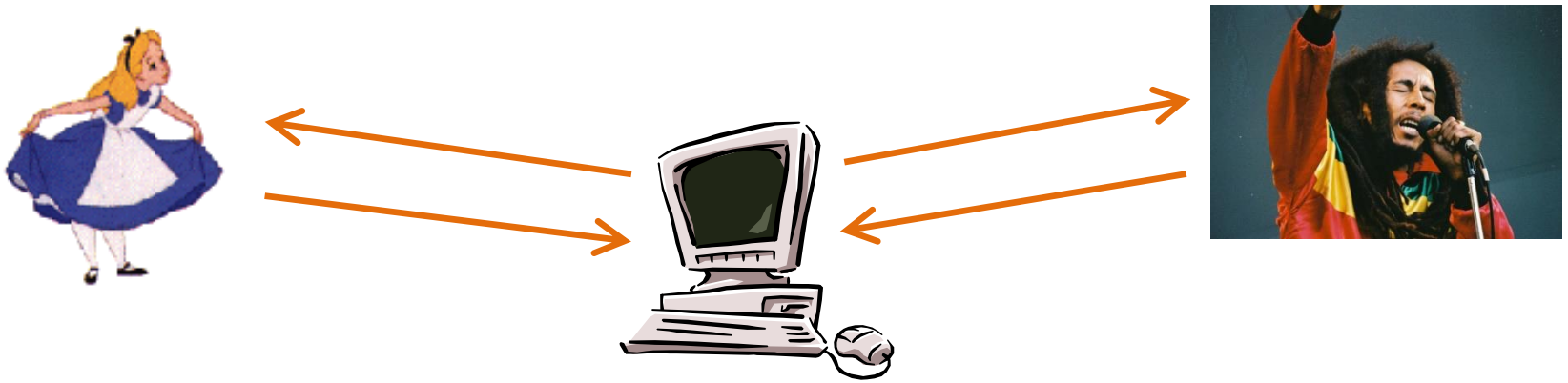
Trivial Quantum Randomness Solution!



Problem: What if your quantum hardware was backdoored by the NSA? (Like the DUAL_EC_DRBG pseudorandom generator was?) Want to trust a deterministic classical computer only

Earlier Approach: Bell-Certified Randomness Generation

Colbeck and Renner, Pironio et al., Vazirani and Vidick, Coudron and Yuen, Miller and Shi…



Upside: Doesn't need a QC; uses only "current technology" (though loophole-free Bell violations are only ~2 years old)

Downside: If you're getting the random bits over the Internet, how do you know Alice and Bob were separated?

Randomness from Quantum Supremacy Experiments

Key Insight: A QC can solve certain sampling problems quickly—but under plausible hardness assumptions, it can **only** do so by sampling (and hence, generating real entropy)

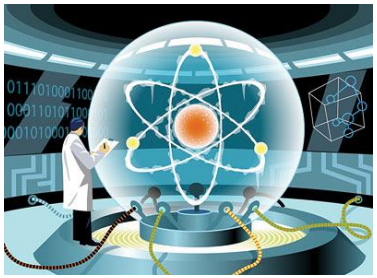
Upsides: Requires just a single device—good for certified randomness over the Internet. Ideal for near-term devices

Caveats: Requires hardness assumptions and initial seed randomness. Verification (with my scheme) takes exp(n) time

Why? Because a classical server could always replace its randomness source by a pseudorandom one without the client being able to detect it

Indeed, our protocol requires certain tasks (e.g., finding heavy outputs of a quantum circuit) to be **easy** for QCs, and other tasks (e.g., finding the same heavy outputs every time) to be **hard** for QCs!

Applications



For the QC owner:
Private randomness

For those connecting over the cloud: Public randomness

The protocol does require pseudorandom challenges, but:

Even if the pseudorandom generator is broken later, the truly random bits will remain safe (“forward secrecy”)

Even if the seed was public, the random bits can be private

The random bits demonstrably weren’t known to *anyone*, even the QC, before it received a challenge (freshness)

The Protocol

1. The classical client generates n-qubit quantum circuits $C_1,\ldots,C_T$ pseudorandomly (mimicking a random ensemble)

2. For each t, the client sends C_t to the server, then demands a response S_t within a very short time

   In the "honest" case, the response is a list of k samples from the output distribution of $C_t|0\rangle^{\otimes n}$

3. The client picks O(1) random iterations t, and for each one, checks whether S_t solves "HOG" (Heavy Output Generation)

4. If these checks pass, then the client feeds $S=\langle S_1,\ldots,S_T\rangle$ into a classical **randomness extractor**, such as GUV (Guruswami-Umans-Vadhan), to get nearly pure random bits

Main Result

Suppose that suitable hardness assumptions hold, and that the server does at most $n^{O(1)}$ quantum computation per iteration. Suppose also that we run the protocol, for $T \leq 2^n$ steps, and the client accepts with probability >½. Then conditioned on the client accepting, the output bits S are $1/\exp(n^{\Omega(1)})$-close in variation distance to a distribution with **min-entropy** $\Omega(Tn)$.

$$H_{\min}(D) := \min_x \log_2 \frac{1}{\Pr_D[x]}$$

Which means: the extractor will output $\Omega(Tn)$ bits that are exponentially close to uniform

Hardest part: show **accumulation** of min-entropy across the T iterations. E.g., rule out that the samples are correlated

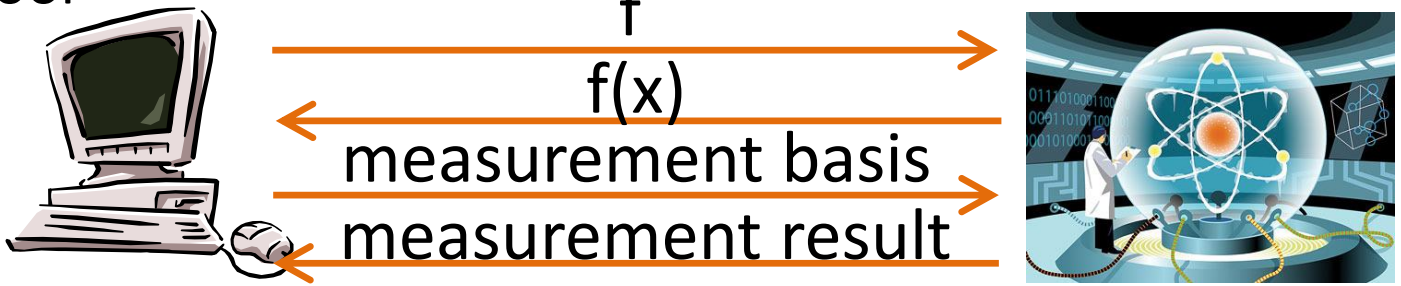
Different Approach

Brakerski, Christiano, Mahadev, Vazirani, Vidick arXiv:1804.00640

Method for a QC to generate random bits, assuming the quantum hardness of breaking lattice-based cryptosystems

2-to-1 function f,
plus trapdoor

$$\sum_x |x\rangle |f(x)\rangle \rightarrow \frac{|x\rangle + |y\rangle}{\sqrt{2}}$$



Huge advantage of the BCMVV scheme over mine:
Polynomial-time classical verification!

Advantage of mine: Can be run on near-term devices!

Future Directions

Can we get quantum supremacy, as well as certified randomness, under more “standard” and less “boutique” complexity assumptions?

Can we get polynomial-time classical verification and near-term implementability at the same time?

Can we get more and more certified randomness by sampling with the **same** circuit C over and over? Would greatly improve the bit rate, remove the need for a PRF

Conclusions

We might be close to ~70-qubit quantum supremacy experiments. We can say nontrivial things about the hardness of simulating these experiments, but we'd like to say more

Certified randomness generation: the **most** plausible application of very-near-term QCs?

This application **requires** sampling problems: problems with definite answers (like factoring) are useless!

Not only can we do it with ~70 qubits, we don't **want** more. No expensive encoding needed; can fully exploit hardware

With this application, all the weaknesses of sampling-based quantum supremacy experiments become strengths!