RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

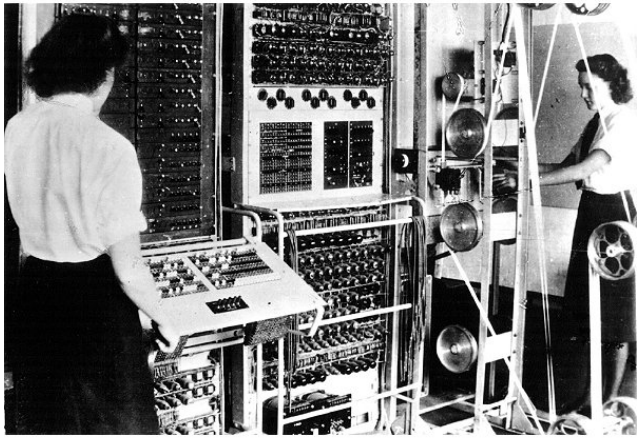
SESSION ID: PDAC-F03

QUANTUM COMPUTING IS HERE, POWERED BY OPEN SOURCE

Konstantinos Karagiannis

Chief Technology Officer
BT Americas
@KonstantHacker

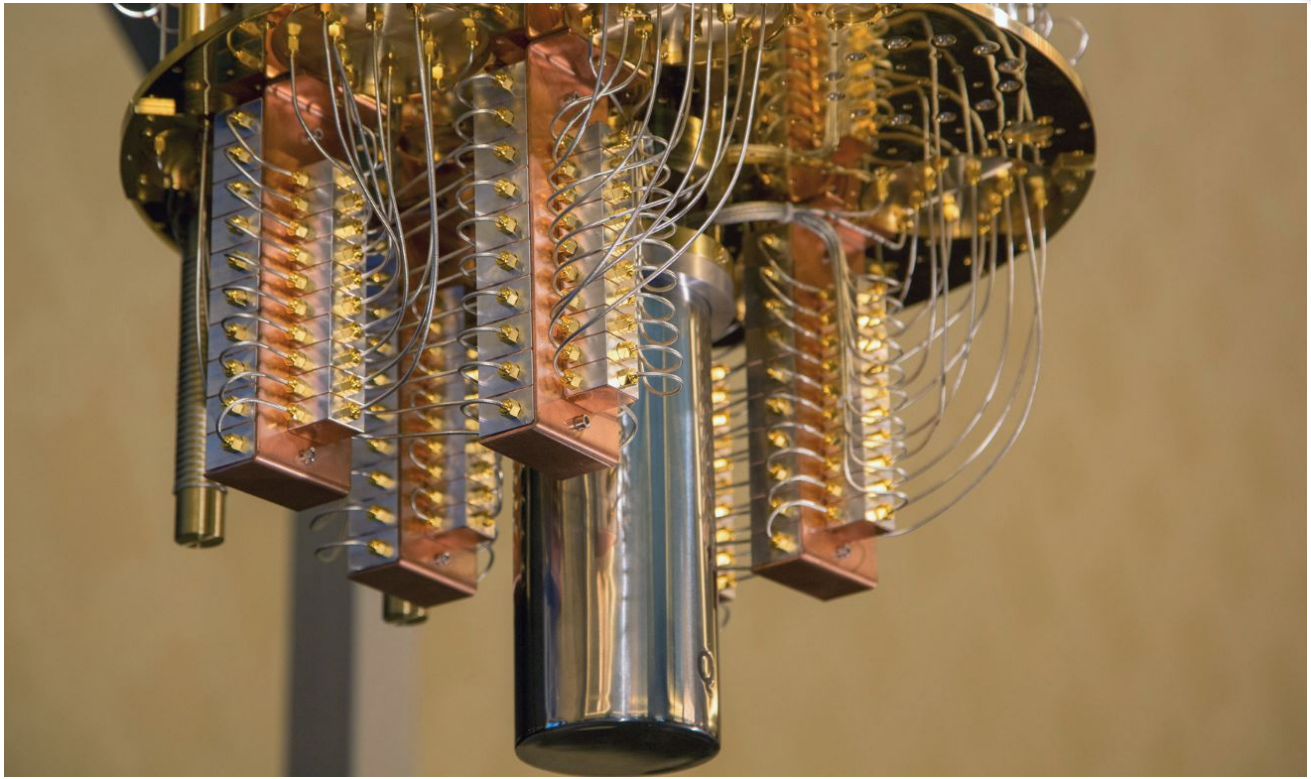
First supercomputer, built to crack encryption



Current supercomputers

As of Nov 2017, all of the top 500 supercomputers run Linux (www.top500.org)

The next gen of supercomputer…

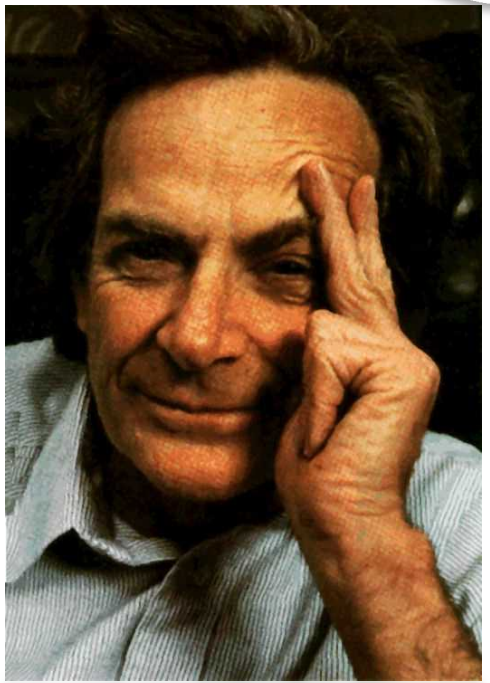


Understanding QM?

“I think I can safely say that no one understands quantum mechanics.” –

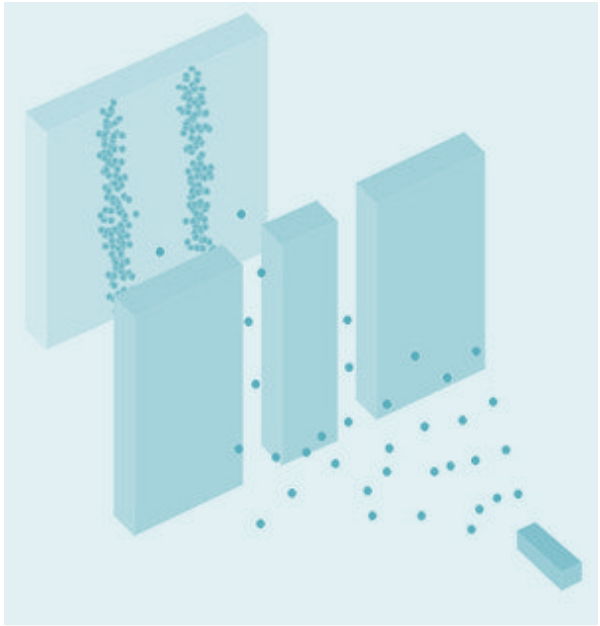
Richard Feynman

(Nobel Prize, 1965)



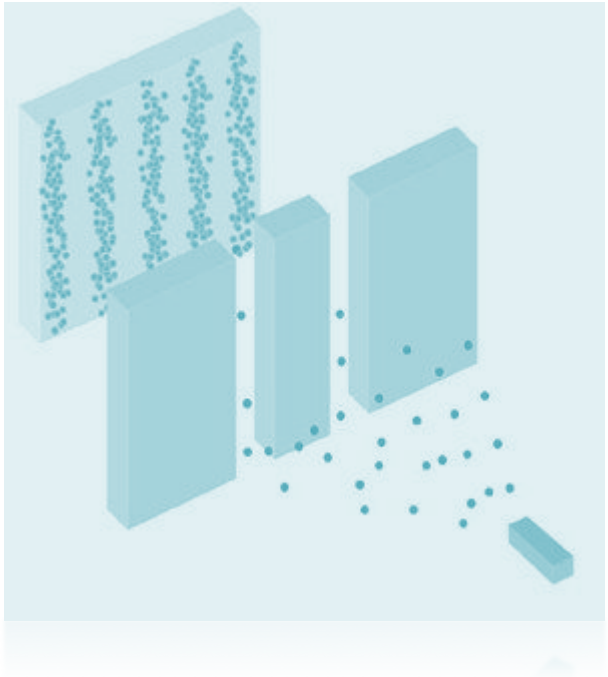
We’ll start with quanta being discrete bundles of energy and give it a shot anyway…

Particle-wave duality



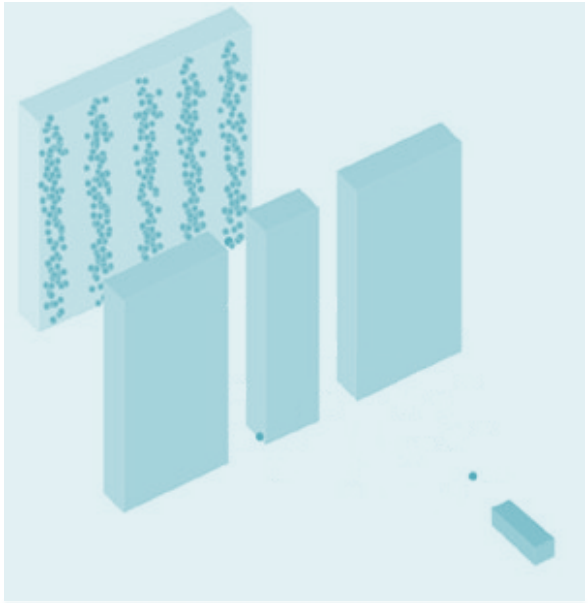
Expected particle behavior or “pooling”

Particle-wave duality



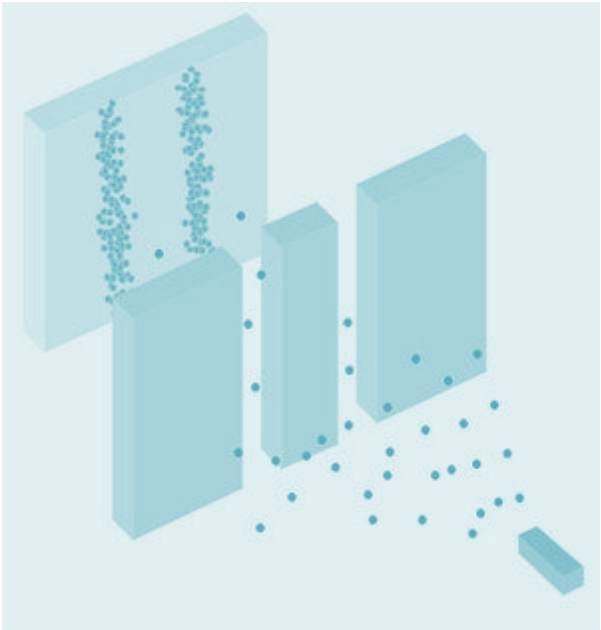
Wave pattern without observation of which slit a particle goes through

Particle-wave duality



Even one particle at a time creates wave pattern

Particle-wave duality



Use a detector on either slit, and pooling appears: particle-wave duality

Superposition

- Observing either slit destroyed quantum superposition
- Decoherence—of critical importance in QC
- Quantum weirdness can't occur on a macro scale because universe makes "observations"
- We can use superposition to make an important element in computing be 2 things at once

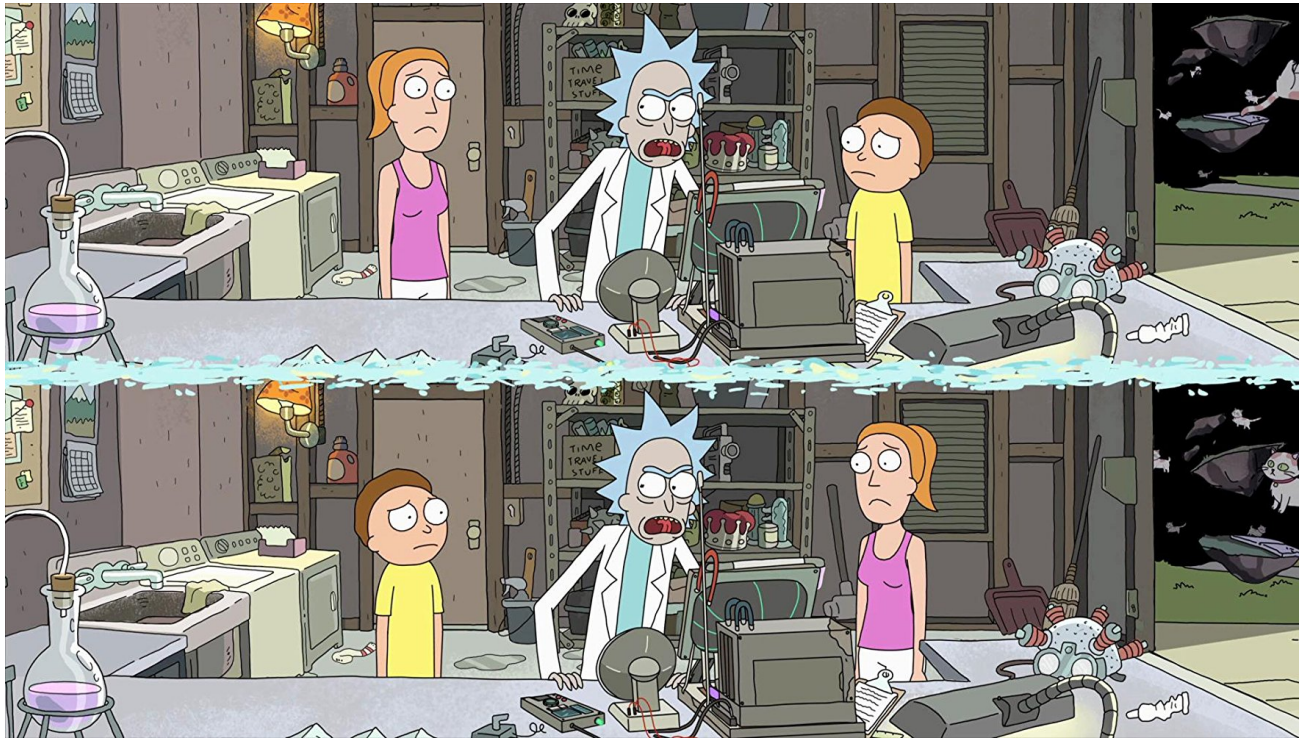
Copenhagen (20s) vs. Many Worlds Interpretation (50s)

Niels Bohr Werner Heisenberg

David Deutsch

Hugh Everett III

Many Worlds Interpretation



MWI gets out of hand quickly

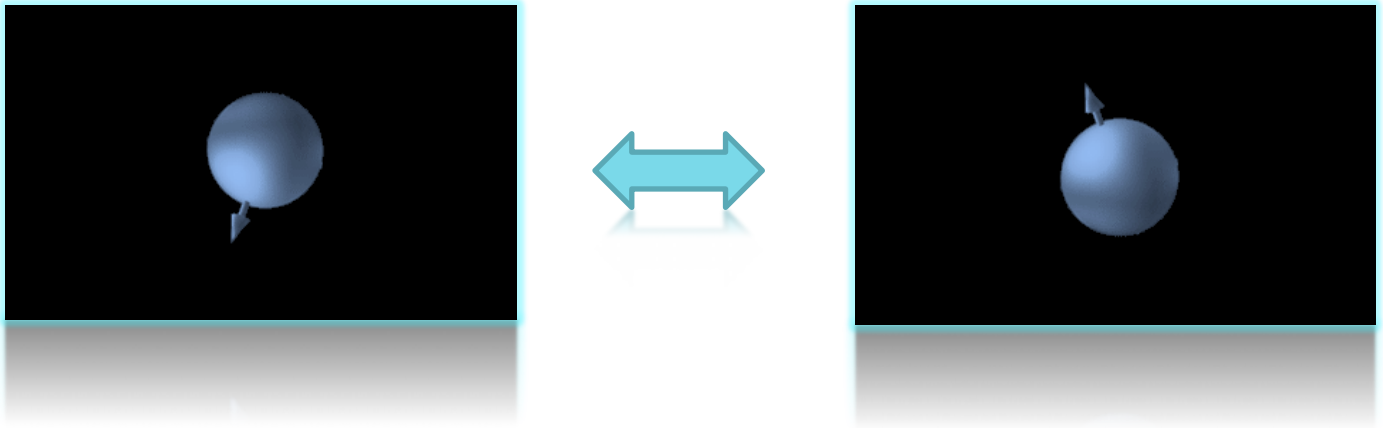
Quantum computer “reports back”

Entanglement

- Back to superposition—Einstein troubled by a type called entanglement
  - “God does not play dice with the universe.”
  - “Spooky action at a distance.”
- Quantum event entangles particles—share quality in superposition (say, spin up/down)
- Until measured/observed, each particle is in both states

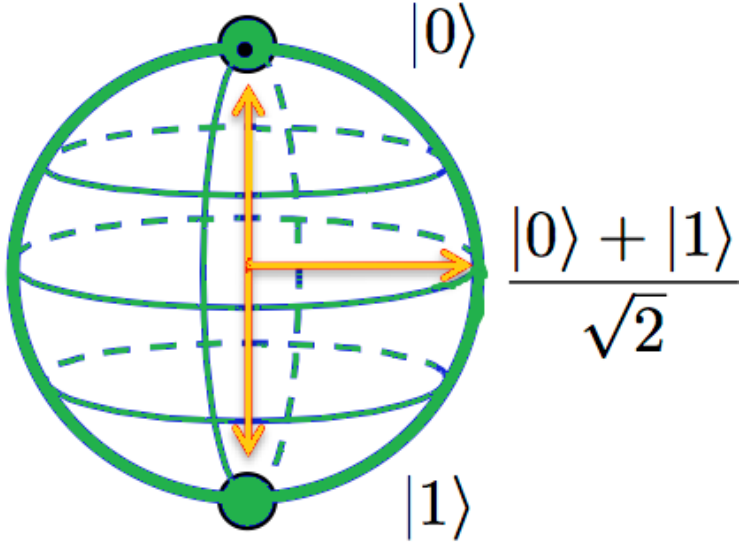
Spooky decoherence

- Observe the spin of one particle, decoherence occurs
- If one particle is spin down, we know instantly the other is spin up
- *Instantly* could mean faster than light!?
- No. Information sent by entanglement is random—can't send 0/1, etc.



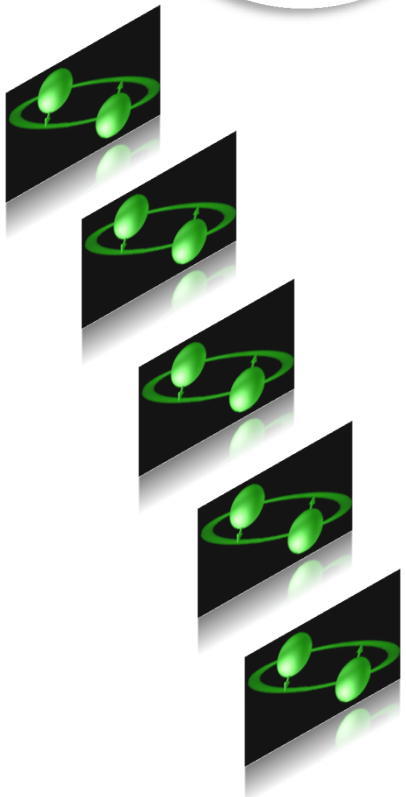
Enter the qubit

- Unlike bits, qubits can be:
  - Zero
  - One
  - Or a superposition of both (with probabilities of each)
- Qubits can perform certain functions with a percentage of effort of a classical computer



Staying coherent

- QCs must maintain coherence in many particles via:
  - Quantum optics
  - Single atom silicon
  - “Large” artificial qubits
  - NMR
  - Discord
- 2012 Nobel Prize for this work:
  - Serge Haroche (France)
  - David Wineland (USA)
- On to what a coherent QC can do…



Cracking public-key crypto

- PK crypto relies on a classical computer's difficulty at factoring large numbers
- Example—find factors of a 400-digit number:

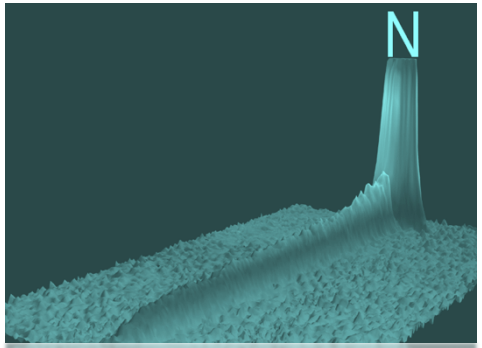
  400-digit number = 200-digit number * 200-digit number

Shor's Algorithm

- 1994—Peter Shor showed a QC could find the factors of large numbers quickly
- Shor's Algorithm has likely answers interfere constructively, unlikely ones destructively
- First proven on a simple QC with four photonic qubits, showing 15=3*5
- Imagine all PK in the clear!
- Risk to Bitcoin because affects elliptic curve cryptography, too

$$\Psi = \frac{1}{2^{n_1}} \sum_{i=0}^{2^{n_1}-1} |\, i >_1 \otimes |\, 0 >_2$$

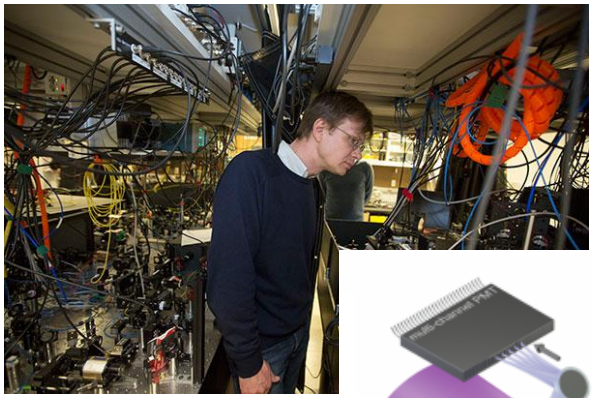
$$\Psi = \frac{1}{2^{n_1}} \sum_{i=0}^{2^{n_1}-1} |\, i >_1 \otimes |\, x^i \mod N >_2$$

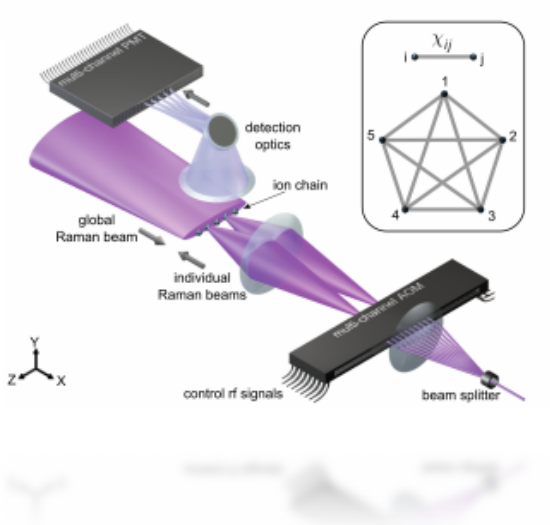


Grover's Algorithm

- Traditional database searches require N/2 searches for N entries
- Peter Grover showed in 1996 how a QC would allow for √N searches
- Could impact DES if encrypted file and source are available—classical computer would need to search 2^{55} keys, but quantum only 185 million
- Proven in lab to work via 2-qubit diamond chip system: 1 search for 4 items instead of 2

Government and university building blocks

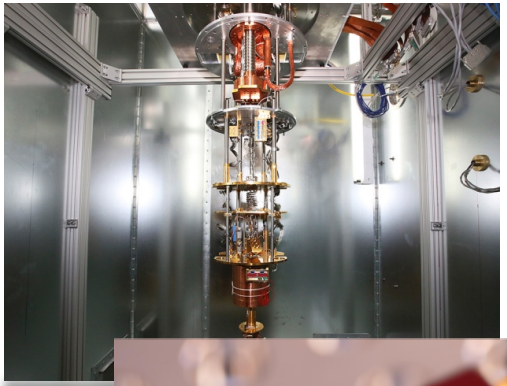


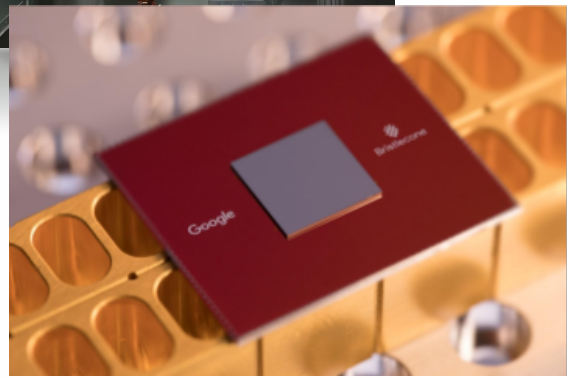


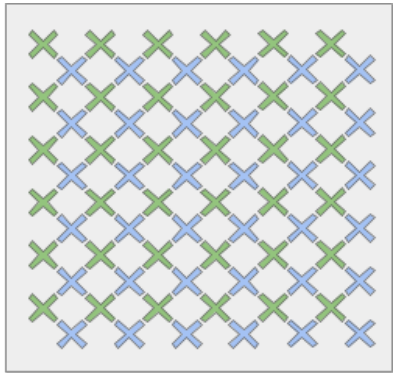
- Harvard University teamed with MIT to build 51-qubit quantum computer (atoms cooled and held by lasers)
- University of New South Wales received first installment of Aus $46 million to build practical quantum computer
- Univ. of Maryland trapped-ion 5-qubit module can run quantum algorithms by executing any sequence of universal logic gates—expected to allow massive chaining of modules
- Univ. of Sussex also announced a modular design

Google searches for quantum search capability

- Google and NASA have a 512-qubit D-Wave at their Quantum Artificial Intelligence Lab
- Google could benefit from Grover's—oddly only mention:
  - Efficient recognizers
  - Polluted data handlers
- Google now working on Bristlecone 72-qubit machine. Seeking "quantum supremacy"
- Qubits not all "connected" (more later)

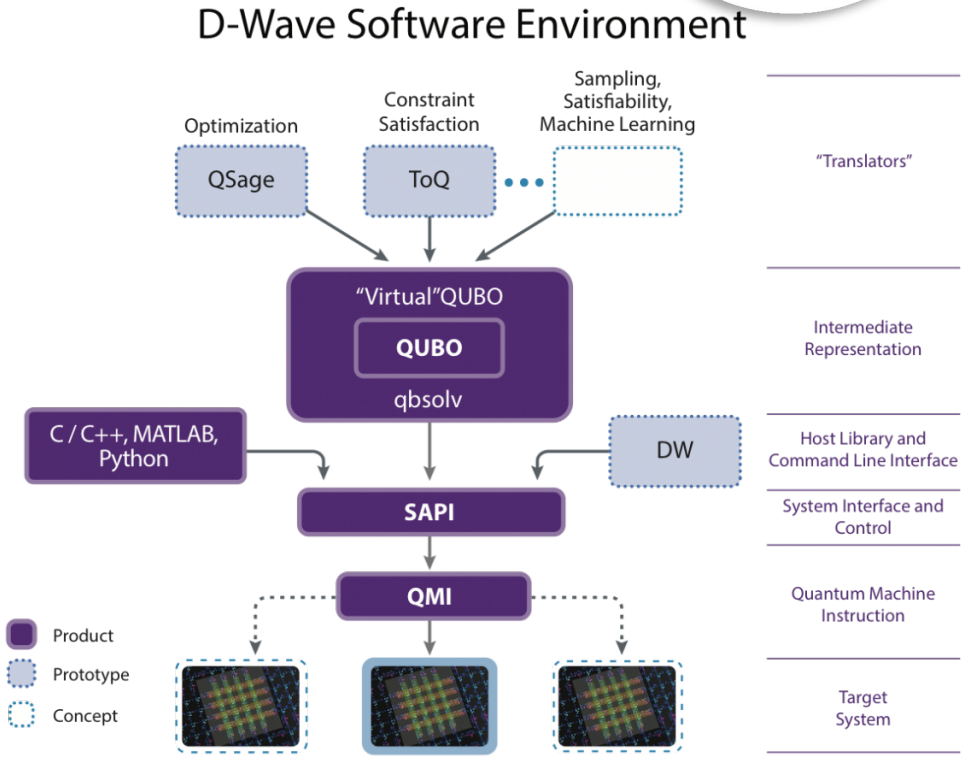




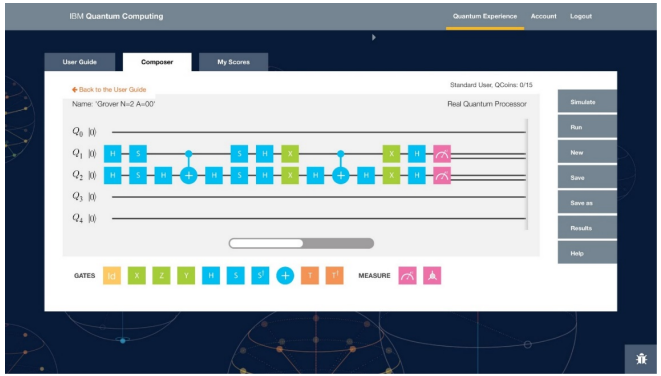


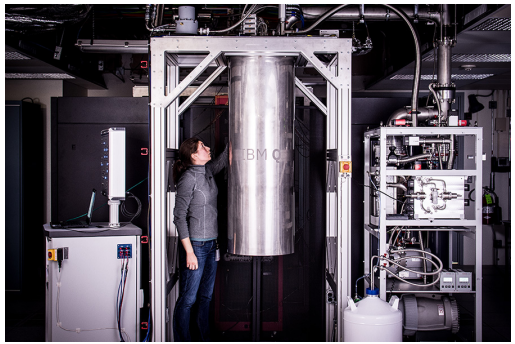
D-Wave and Qbsolv

- Despite not providing a “universal” QC, D-Wave has open source dev environment for its machines (or D-Wave simulator software to test)
- Internet API (RESTful), with C/C++, Python, and MATLAB client libraries
- Can be set up as cloud resource or in high-performance computing environment. Also available via D-Wave’s hosted cloud service.
- D-Wave dev tools and client libraries allow developers to create algorithms and applications



IBM

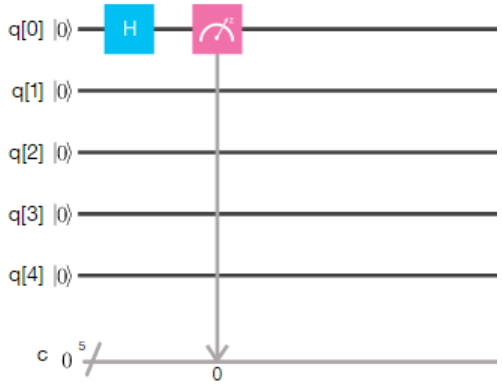




- IBM made headlines by letting you try out a working 5-qubit system—the Quantum Experience Chip. Upped to 16 qubits for free (http://www.research.ibm.com/quantum/)
- You can now register to use a 20-qubit chip! (To compare, Alibaba announced 11 qubits)
- You can run Grover's search live and see the future
- IBM betting big on quantum—50-qubit (offline) machine

QISKit—programming real quantum computers

- In addition to using a visual Composer on the site, there are open dev kits available for the IBM Quantum Experience Chip
- QISKit https://github.com/QISKit Python SDK allows:
  - Building of quantum circuits that represent a problem
  - Compiling to run on different backends (simulators/real chips of different quantum volumes)
  - Running the jobs
- ProjectQ lets users also simulate quantum programs on classical computers, emulating at a higher level of abstraction



ProjectQ and getting started with quantum coding

- Getting started with ProjectQ is pretty simple:

```
python -m pip install --user projectq
```

- Syntax well explained at http://projectq.readthedocs.io
- Similar to syntax in quantum physics:

$R_x(\theta)|\text{qubit}\rangle$

becomes

```
Rx(theta) | qubit
```


```
from projectq import MainEngine  # import the main compiler engine
from projectq.ops import H, Measure  # import the operations we want to perform (Hadamard

eng = MainEngine()  # create a default compiler (the back-end is a simulator)
qubit = eng.allocate_qubit()  # allocate 1 qubit

H | qubit  # apply a Hadamard gate
Measure | qubit  # measure the qubit

eng.flush()  # flush all gates (and execute measurements)
print("Measured {}".format(int(qubit)))  # output measurement result
```

Demo code lets you get qubits spinning on IBM QEC

ProjectQ Demo

Compiling code for IBM QE

Import the IBM setup, the gates, and the compiler engine:

In [1]:
```
import projectq.setups.ibm # Imports the default compiler to map to IBM QE
from projectq.backends import IBMBackend
from projectq.ops import Measure, Entangle
from projectq import MainEngine
```

Create the compiler using the default compiler engines for the IBM backend and allocate a quantum register of 3 qubits:

In [2]:
```
engine = MainEngine(IBMBackend(use_hardware=True, num_runs=1024, verbose=True))
qureg = engine.allocate_qureg(3)
```

Entangle the quantum register:

In [3]:
```
Entangle | qureg
```

Measure the quantum register and run the circuit:

In [4]:
```
Measure | qureg
engine.flush()
```

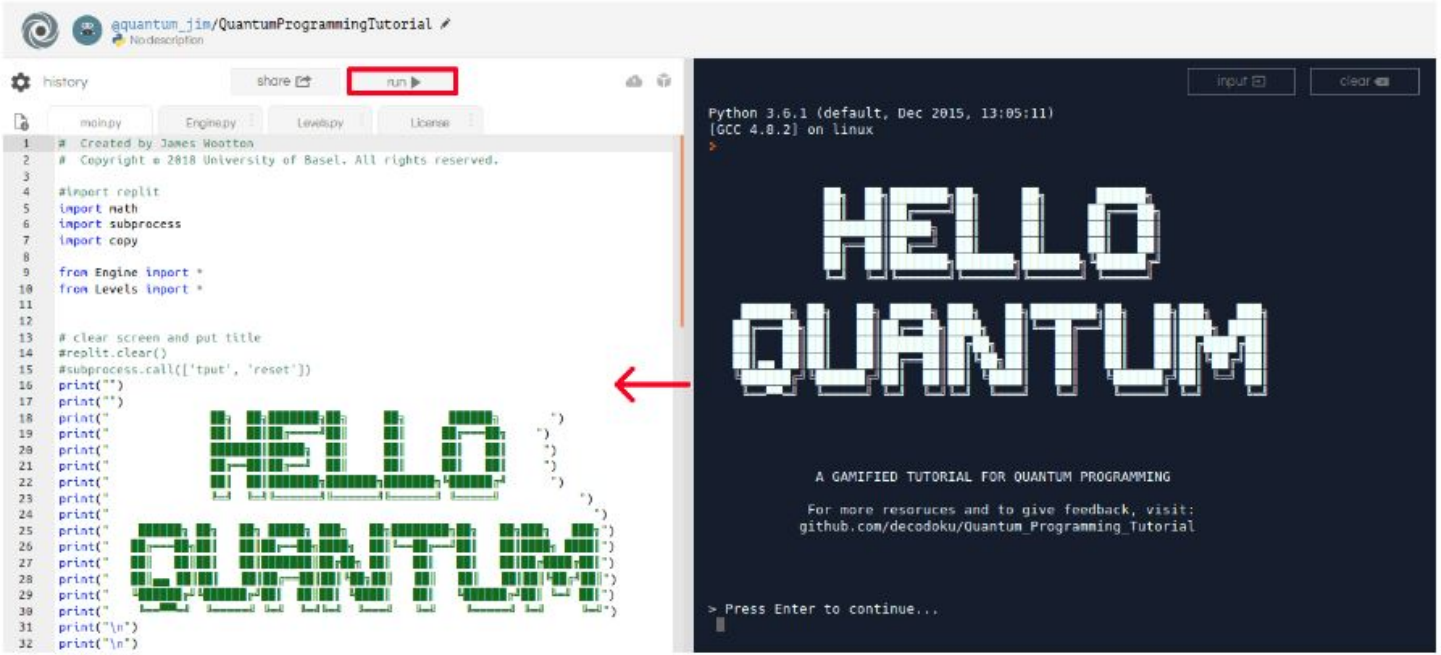
```
Authenticating...
IBM QE user (e-mail) > dsteiger@phys.ethz.ch
IBM QE password > ........
Running code...
Waiting for results...
Done.
00000 with p = 0.4677734375*
10000 with p = 0.00390625
01000 with p = 0.009765625
11000 with p = 0.0224609375
00100 with p = 0.005859375
10100 with p = 0.0263671875
01100 with p = 0.0322265625
11100 with p = 0.431640625
```

Output the measurement result:

In [5]:
```
print([int(q) for q in qureg])
```

```
[0, 0, 0]
```

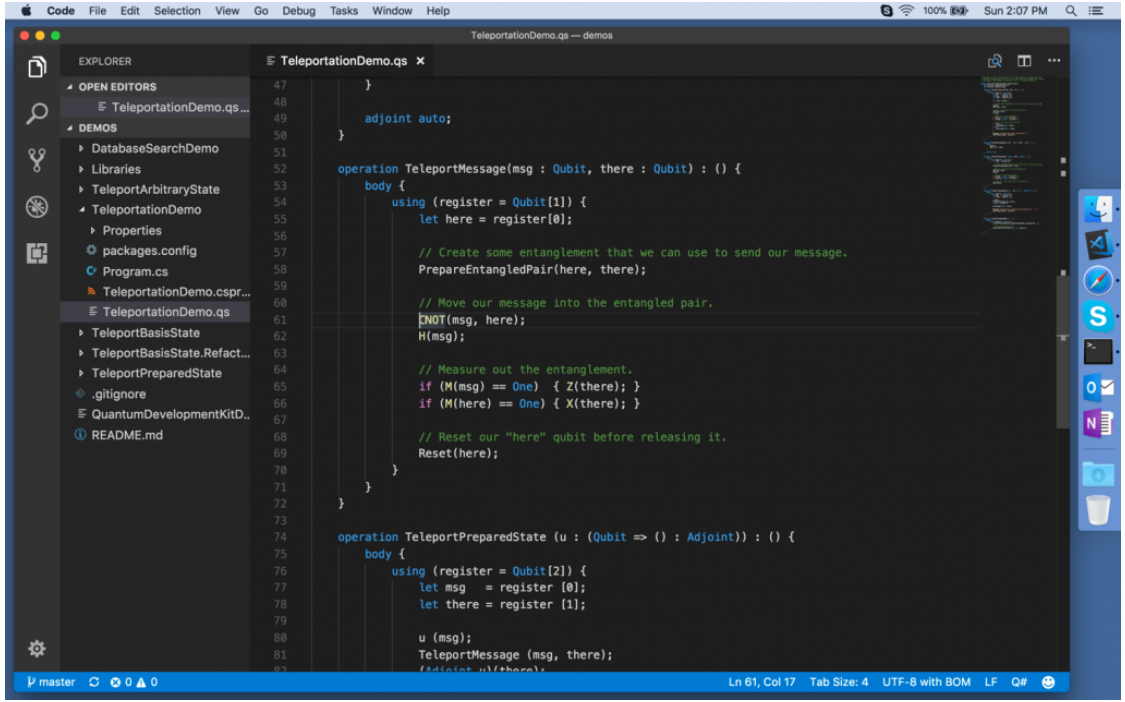
Games make learning fun



https://github.com/decodoku/Quantum_Programming_Tutorial/

Recall “connections” when going from simulator to real

Microsoft Quantum Development Kit



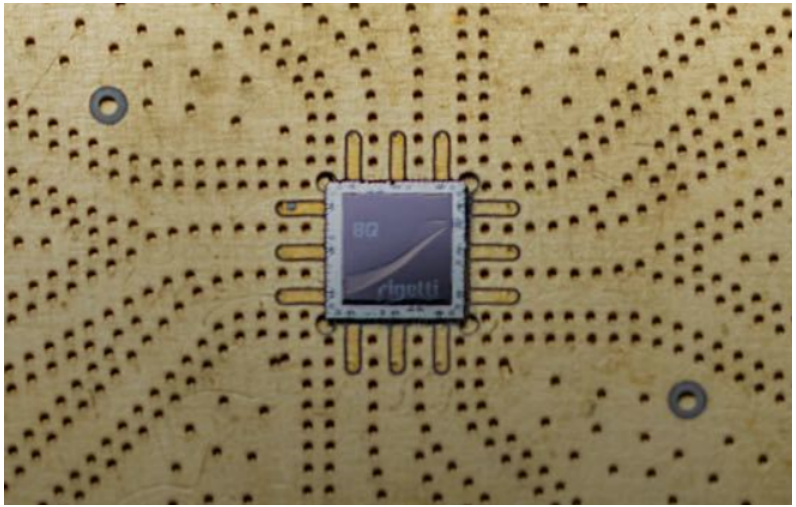
- Introduced Q# programming language
- Supports Windows, Mac OS and Linux (!)
- Open source libraries at https://github.com/microsoft/quantum
- Runs on a quantum platform ***simulator***, locally or cloud,so some criticisms
- MS developing a quantum computer based on topological qubits. Majorana FTW?

Rigetti Forest

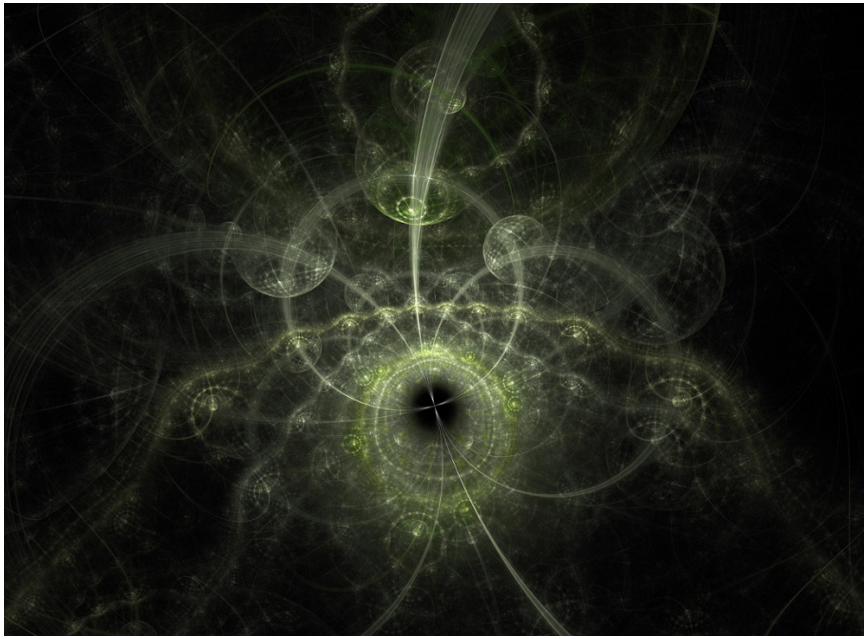
- Private company with a 19-qubit processor
- Forest API/developer environment for creating quantum software
- QUIL open quantum instruction language based on shared classical/quantum model
- Example Hadamard gate (Shor)

```
DEFGATE HADAMARD:
    1/sqrt(2), 1/sqrt(2)
    1/sqrt(2), -1/sqrt(2)
```

- Free cloud access to 26 simulated qubits



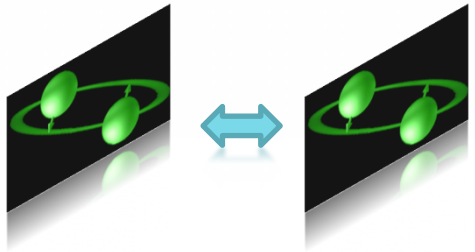
Post-quantum encryption



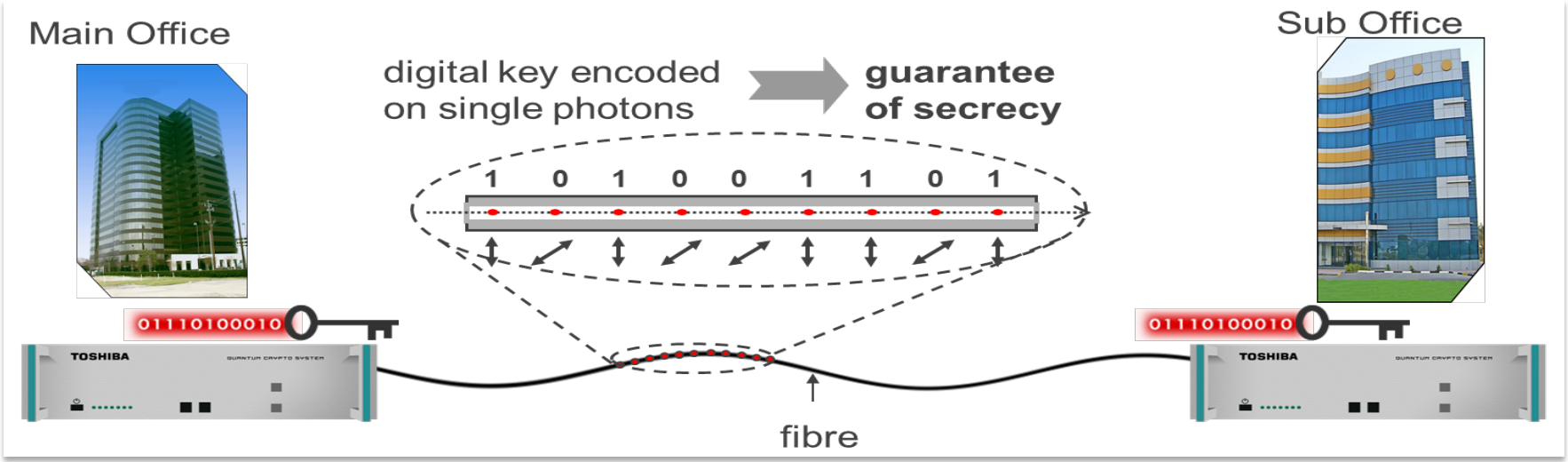
- Shor's proven in lab against PK—RSA, Diffie-Hellman, elliptic curve (blockchain!)
- Grover aids brute force (AES-256 = AES-128)
- The following still safe:
  - Code based
  - Hash based
  - Lattice based
  - Multivariate quadratic equations
  - One time pad (QKD)

Quantum keys to the future (for some use cases)

- A point-to-point protection from the quantum threat
- Quantum encoded keys sent via a relevant medium—e.g., photons via fiber
- One-time pad system protected by QKD
- Any attempt to tap signal can be detected and prevented
- Works today, provides hope until **complete quantum systems** appear
- qBitcoin proposes a cryptocurrency sent this way (new networks needed, though)



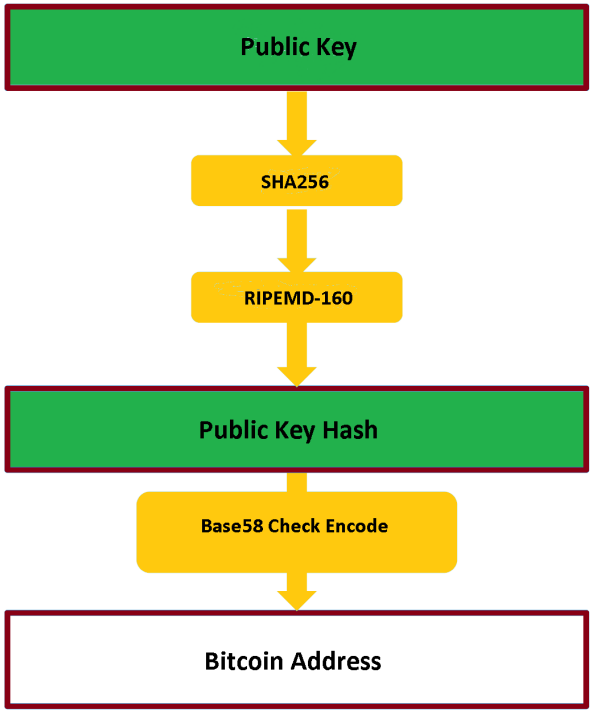
Quantum channel on shared fiber with 200Gbps bidirectional traffic (9/15)



Open Quantum Safe

- Despite QKD's effectiveness, it's not a "use everywhere" solution
- Promising new quantum encryption: liboqs (https://openquantumsafe.org/)
- Open source C library for quantum-resistant crypto algorithms
- API suitable for post-quantum key exchange algorithms
- Open-quantum-safe/openssl is integration of liboqs into OpenSSL 1.0.2. Goal of providing easy prototyping of quantum-resistant cryptography

What about ECC and "fixing" open source money?



- Bitcoin wallet addresses made of: Public key, private key, and address
- Public key derived from private key by elliptic curve multiplication
- Address derived by:
  - applying SHA256 hash function to public key
  - applying RIPEMD-160 hash function
  - adding checksum for error correction
- "Used" bitcoin or other entities have public keys exposed on blockchain—can be downloaded, cracked offline

Fixing Bitcoin and Ethereum?

- Bitcoin is in the biggest bind. Lamport signatures may be able to help.
- Lamport public key consists of 320 hashes rather than an elliptic curve point—address is SHA256+RIPEMD-160 hash of public key
- Even with Grover's algorithm, it takes 2^80 steps to construct a fraudulent transaction or 2^80 * 80 steps to crack all hashes (trillions of trillions)
- Ethereum roadmap already includes plans to gradually become post-quantum ready, beginning with wallets

Quantum Resistant Ledger

- What about entirely new blockchains, built to resist?
- A promising post-quantum blockchain is the Quantum Resistant Ledger http://theqrl.org/
- The Winternitz OTS+ and Extended Merkle Signature Scheme (XMSS)
- Proof of Stake (Ethereum moving to this too)

Getting ready for the revolution

- Most exciting fact: working quantum computers are here today, and you can access them (or simulations) for free with open source
- In true open source fashion, you can contribute, too (QISKit has over 50 serious contributions, including papers)
- Plan on spending the next 3 years learning how to master these machines as they grow in power
- Plan on spending the next 3 years also working on post-quantum crypto in your environment. The crypto threat is real!

THANKS

@KonstantHacker