# Quantum Attacks on Bitcoin and Countermeasures

Gavin K Brennen

Dept. of Physics

Macquarie University, Sydney

Divesh Aggarwal

Troy Lee (CQT)

Miklos Santha

Marco Tomamichel (UTS)

arXiv:1710.10377

MACQUARIE University

EQUS

Australian Government

Australian Research Council

BITCOIN SECURITY OCTOBER 16, 2016 18:58

# Quantum Computers Will Destroy Bitcoin, Scientists Warn


CRYPTO ECONOMY

## Stealing Bitcoin with Math

**Author:** David Armitage - **Date:** 17 Oct 2017

Quantum computing is advancing so fast that bitcoin could become hackable within months, not years. The NSA

# The Coming Bitcoin Bust

BANYAN HILL
AMERICA'S NO.1 SOURCE FOR SMARTER, SAFER AND MORE PROFITABLE INVESTING

Currency October 24, 2017 Jeff Yastine


It was the sound of digital money being created as each machine's chip strained to solve another piece of an elaborate cryptographic puzzle — the very basis for the cryptocurrency — and unlock just a little more bitcoin.

It was all gone now. The room had a funereal silence.

The great cryptocurrency boom had gone bust.

No one could explain why, at first. Bitcoin mysteriously plummeted in value for weeks, then months. But news leaks had finally identified the root of the problem…


"Damned quantum computers," muttered the man as he shut off the lights and walked out for the last time.

# Oct. 31, 2008: 9 page whitepaper posted on online forum

# Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

**Abstract.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Jan. 9 2009 Version 0.1 of bitcoin software on Sourceforge

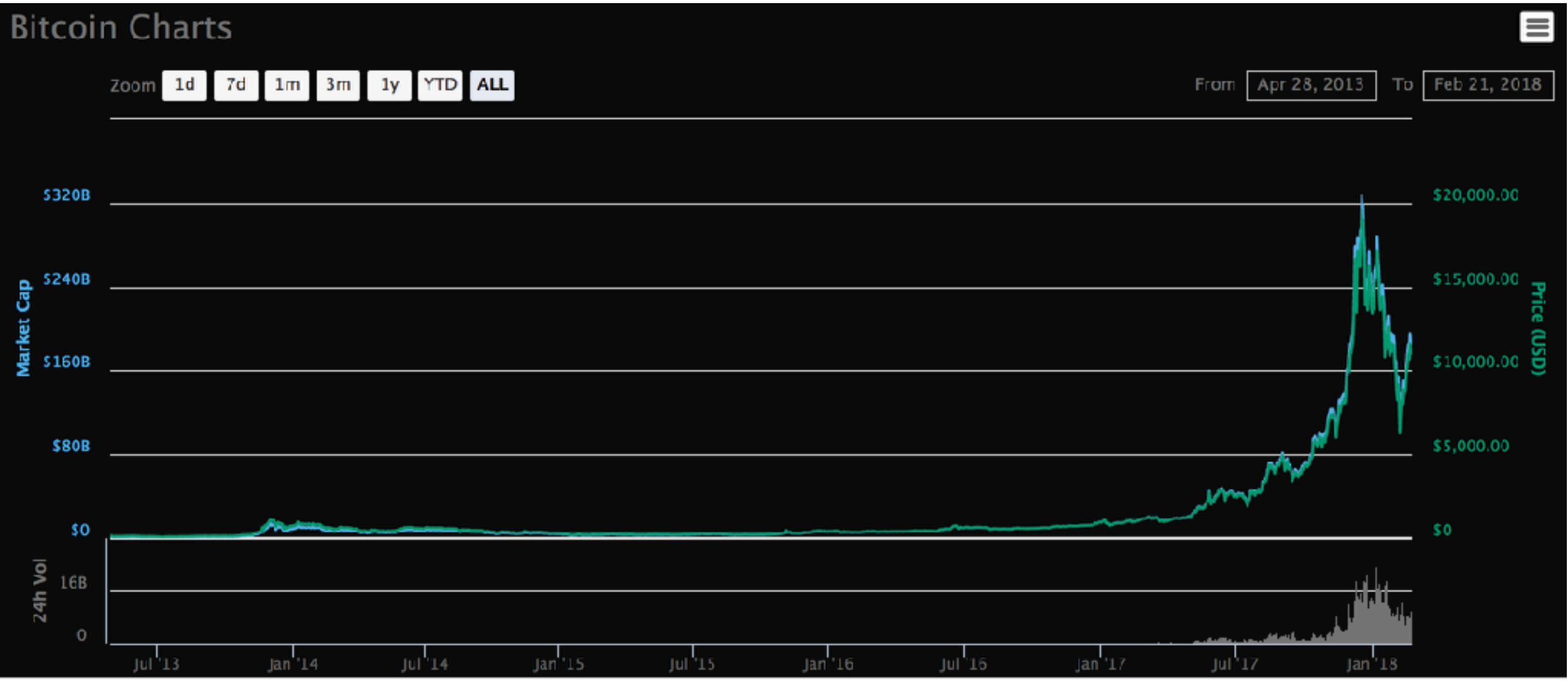

Elon Musk denies he is bitcoin creator Satoshi Nakamoto

Arjun Kharpal | @ArjunKharpal

Published 8:08 AM ET Tue, 28 Nov 2017 | Updated 7:07 PM ET Tue, 28 Nov 2017

CNBC

Today coinmarketcap.com




**Cryptocurrencies: 1515 / Markets: 8579** **Market Cap: $477,421,081,182 / 24h Vol: $26,016,663,341 / BTC Dominance: 39.3%**

# Cryptocurrency Market Capitalizations

# Prehistory: Hashcash


"Pricing via Processing or Combating Junk Mail," Cynthia Dwork and Moni Naor, presented at Crypto'92

# Hashcash

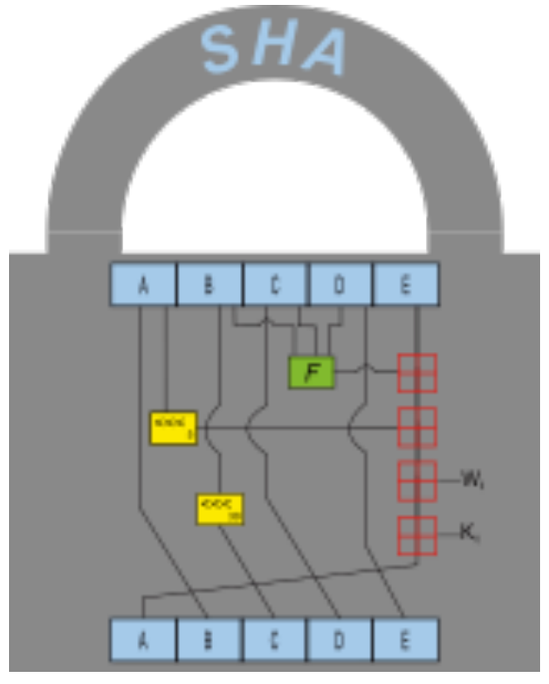
- Idea:  Make it costly to send email by requiring sender solve a hard problem
- Proof of work: Requires much computational effort to solve a problem but the proof can be verified efficiently
- For email: Encoding of a hashcash stamp is added to the header of an email to prove the sender has expended CPU time calculating the stamp prior to sending. Receiver can efficiently verify that the stamp is valid
- Header: 1:20:1303030600:adam@cypherspace.org::McMybZIhxKXu57jd:ckvi

  - *ver*: Hashcash format version, 1 (which supersedes version 0).
  - *bits*: Number of "partial pre-image" (zero) bits in the hashed code.
  - *date*: The time that the message was sent, in the format `YYMMDD[hhmm[ss]]`.
  - *resource*: Resource data string being transmitted, e.g., *recipients* IP address or email address.
  - *ext*: Extension (optional; ignored in version 1).
  - r*and*: String of random characters, encoded in base-64 format.
  - *counter*: Binary counter (up to $2^{20}$), encoded in base-64 format.

# Hashcash Proof-of-work

- Header: 1:20:1303030600:adam@cypherspace.org::McMybZIhxKXu57jd:ckvi
- Sender prepares a header and appends a counter value initialized to a random number. It then computes the 160-bit SHA-1 hash of the header. If the first 20 bits (=5 most significant hex digits) of the hash are all zeros, then this is an acceptable header. If not, then the sender increments the counter and tries the hash again. Out of $2^{160}$ possible hash values, there are $2^{140}$ hash values that satisfy this criterion. Thus the chance of randomly selecting a header that will have 20 zeros as the beginning of the hash is $2^{-20}$. Not too onerous for 1 email, but costly for spammers.
- Receiver calculates the 160-bit SHA-1 hash of the entire string (e.g., `"1:20:060408:adam@cypherspace.org::1QTjaYd7niiQA/sc:ePa"`). This takes about two microseconds on a 1 GHz machine. If the first 20 bits are not all zero, the hash is invalid. (Can adjust the number of zero bits required as computers speed up.). The recipient's computer inserts the hash string into a database. If the string is already in the database (indicating that an attempt is being made to re-use the hash string), it is invalid.

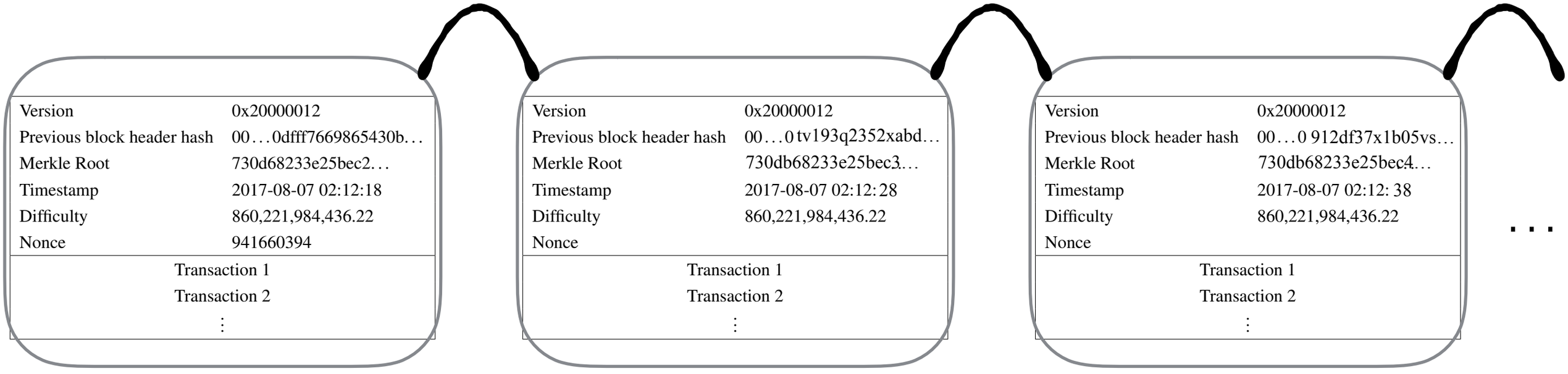
# SHA Hash function




- SHA1: input up to $2^{64}$ bit message, output 160 bit message digest
- High entropy non linear function
  - SHA1("The quick brown fox jumps over the lazy dog")= 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12 (hexadecimal)
  - SHA1("The quick brown fox jumps over the lazy cog")= de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3 (hexadecimal)
- Collisions should to be rare
  - time to find two inputs with same output $O(2^{160/2})$, same as birthday problem
- Yet cryptographers found algorithms to produce collisions faster
- People now use SHA256 with better security. Input up to $2^{64}$ bit message, output 256 bit message digest

# Bitcoin Protocol

- Takes the idea of Hashcash and applies it to digital currency.
- Makes it expensive to legitimate a new transaction which provides security against double spending (sending the same cryptocurrency to two recipients)
- This is done using not a single isolated header stamp but a chain of blocks each with a header
- Tampering with past blocks is detectable in future blocks



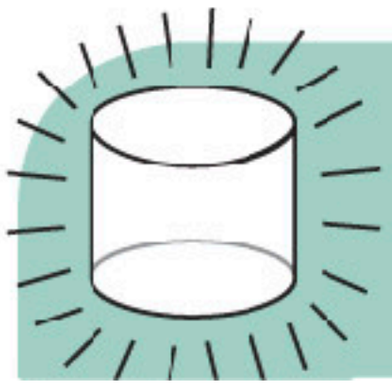
# Anatomy of a Bitcoin transaction

## WALLETS AND ADDRESSES

Bob and Alice both have Bitcoin "wallets" on their computers.

Wallets are files that provide access to multiple Bitcoin addresses.

An address is a string of letters and numbers, such as 1HULMwZEPkjEPeCh43BeKJL1ybLCWrfDpN.



Bob creates a new Bitcoin address for Alice to send her payment to.

## CREATING A NEW ADDRESS

Each address has its own balance of bitcoins.

JOSHUA J. ROMERO, BRANDON PALACIO & KARLSSONWILKER INC.

## SUBMITTING A PAYMENT

Alice tells her Bitcoin client that she'd like to transfer the purchase amount to Bob's address.


Alice's wallet holds the private key for each of her addresses. The Bitcoin client signs her transaction request with the private key of the address she's transferring bitcoins from.

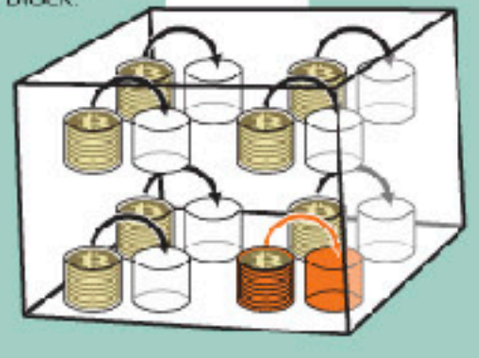
Anyone on the network can now use the public key to verify that the transaction request is actually coming from the legitimate account owner.

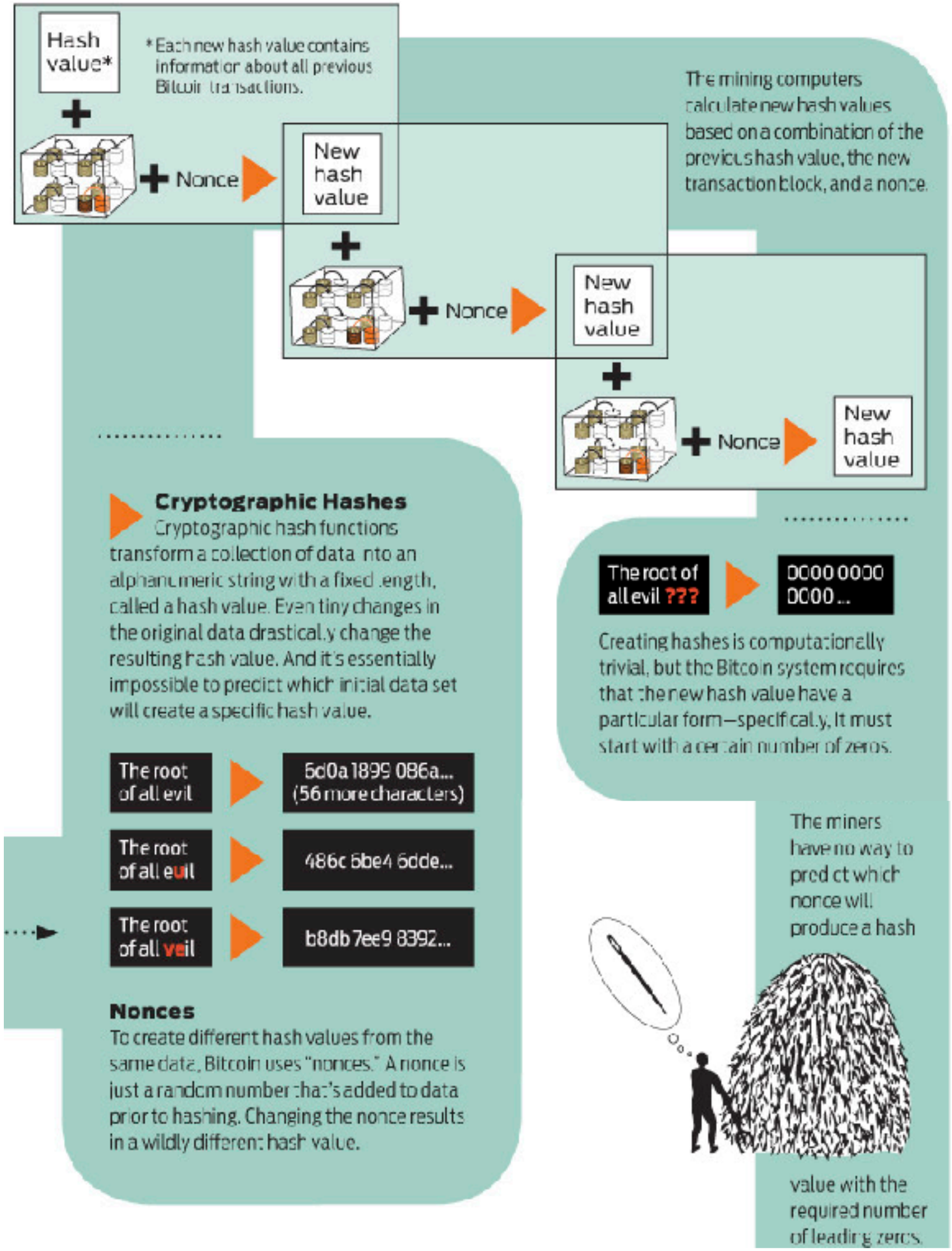
## VERIFYING THE TRANSACTION

Gary, Garth, and Glenn are Bitcoin miners.


Their computers bundle the transactions of the past 10 minutes into a new "transaction block."

The miners' computers are set up to calculate cryptographic hash functions.

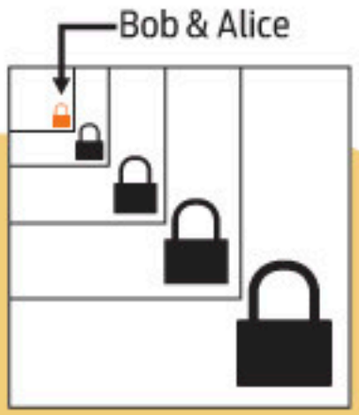



Hash value*
*Each new hash value contains information about all previous Bitcoin transactions.
Nonce
New hash value
Nonce
New hash value
Nonce
New hash value
The mining computers calculate new hash values based on a combination of the previous hash value, the new transaction block, and a nonce.
Cryptographic Hashes
Cryptographic hash functions transform a collection of data into an alphanumeric string with a fixed length, called a hash value. Even tiny changes in the original data drastically change the resulting hash value. And it's essentially impossible to predict which initial data set will create a specific hash value.
The root of all evil
6d0a 1899 086a... (56 more characters)
The root of all evil
486c 6be4 6dde...
The root of all veil
b8db 7ee9 8392...
Nonces
To create different hash values from the same data, Bitcoin uses "nonces." A nonce is just a random number that's added to data prior to hashing. Changing the nonce results in a wildly different hash value.
The root of all evil ???
0000 0000 0000...
Creating hashes is computationally trivial, but the Bitcoin system requires that the new hash value have a particular form—specifically, it must start with a certain number of zeros.
The miners have no way to predict which nonce will produce a hash
value with the required number of leading zeros.

Each block includes a "coinbase" transaction that pays out 50 bitcoins to the winning miner—in this case, Gary. A new address is created in Gary's wallet with a balance of newly minted bitcoins.


## TRANSACTION VERIFIED

As time goes on, Alice's transfer to Bob gets buried beneath other, more recent transactions. For anyone to modify the details, he would have to redo the work that Gary did—because any changes require a completely different winning nonce—and then redo the work of all the subsequent miners. Such a feat is nearly impossible.



# Enter Quantum Computers

- Two threats
- 1. Quantum Computers dominate mining operations, finding solutions to inverse Hash problem much faster and thereby double spending + getting all rewards for verification
- 2. Quantum Computers crack digital signature encryption and spend money from your account without permission
- How dangerous are these attacks and what's the forecasted timeline for a real threat?

# Quantum attacks on mining

- Bitcoin Proof of Work task is to find a block header with sufficient leading 0s

$$h(\text{header}) \leq t$$

  - where the Hashing is applied twice

$$h(\cdot) = \text{SHA256}(\text{SHA256}(\cdot))$$

- Classically, the number of runs of trial headers is $D \times 2^{32}$
  - where the difficulty is $D = 2^{224}/t$
  - as of today D=3 x $10^{12}$

- Hashing has no known structure so the best we could expect is a square root speedup with a quantum computer. This is still a big deal!

# Quantum attacks on mining

- Grover Algorithm
- Let N= $2^{256}$. With probability at least 0.9999 a random set of 10N/t block headers with contain at least one element satisfying $h(\text{header}) \leq t$
- Fix some deterministic, easily computed function g mapping

g: $S = \{0,1\}^{\lceil \log(10 \cdot N/t) \rceil}$ ——> Block headers

- Define indicator function f

$$f(x) = \begin{cases} 0 & \text{if} \quad h(g(x)) > t \\ 1 & \text{if} \quad h(g(x)) \leq t \end{cases}$$

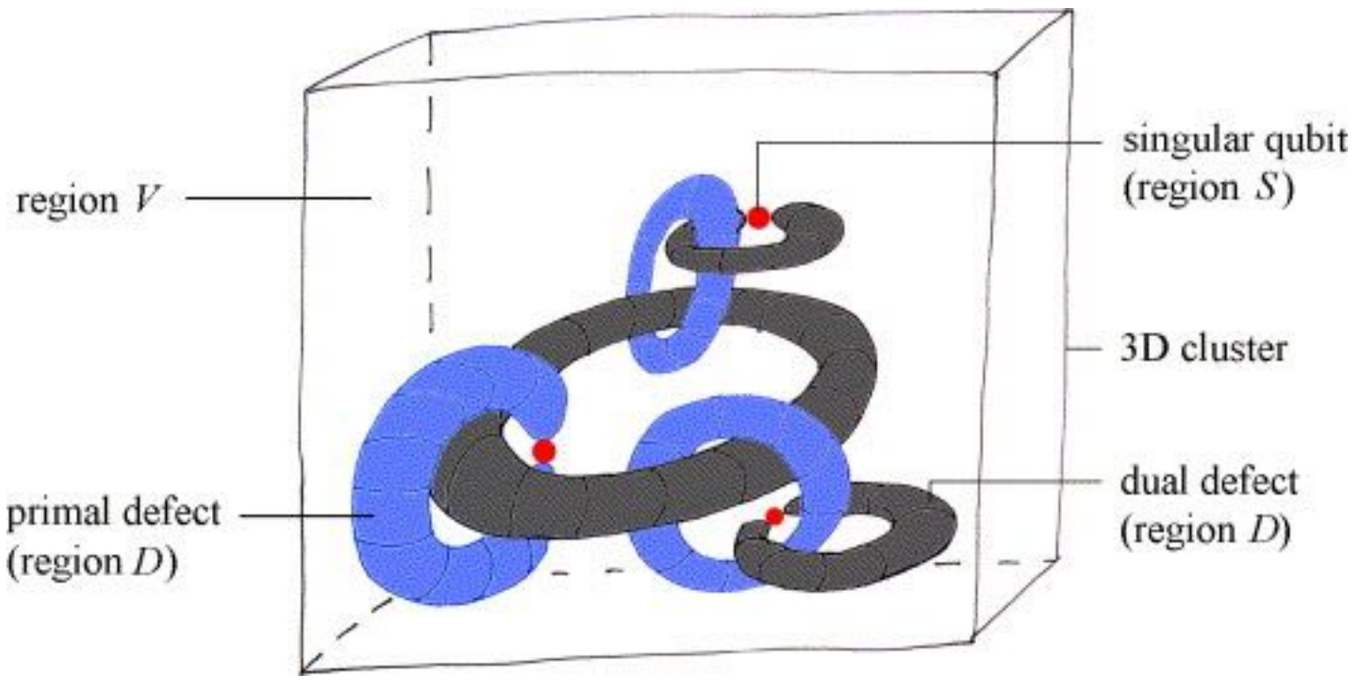
- Run Oracle call on a superposition of all inputs

$$\sum_{x \in S} \alpha_x |x\rangle \rightarrow \sum_{x \in S} (-1)^{f(x)} \alpha_x |x\rangle$$

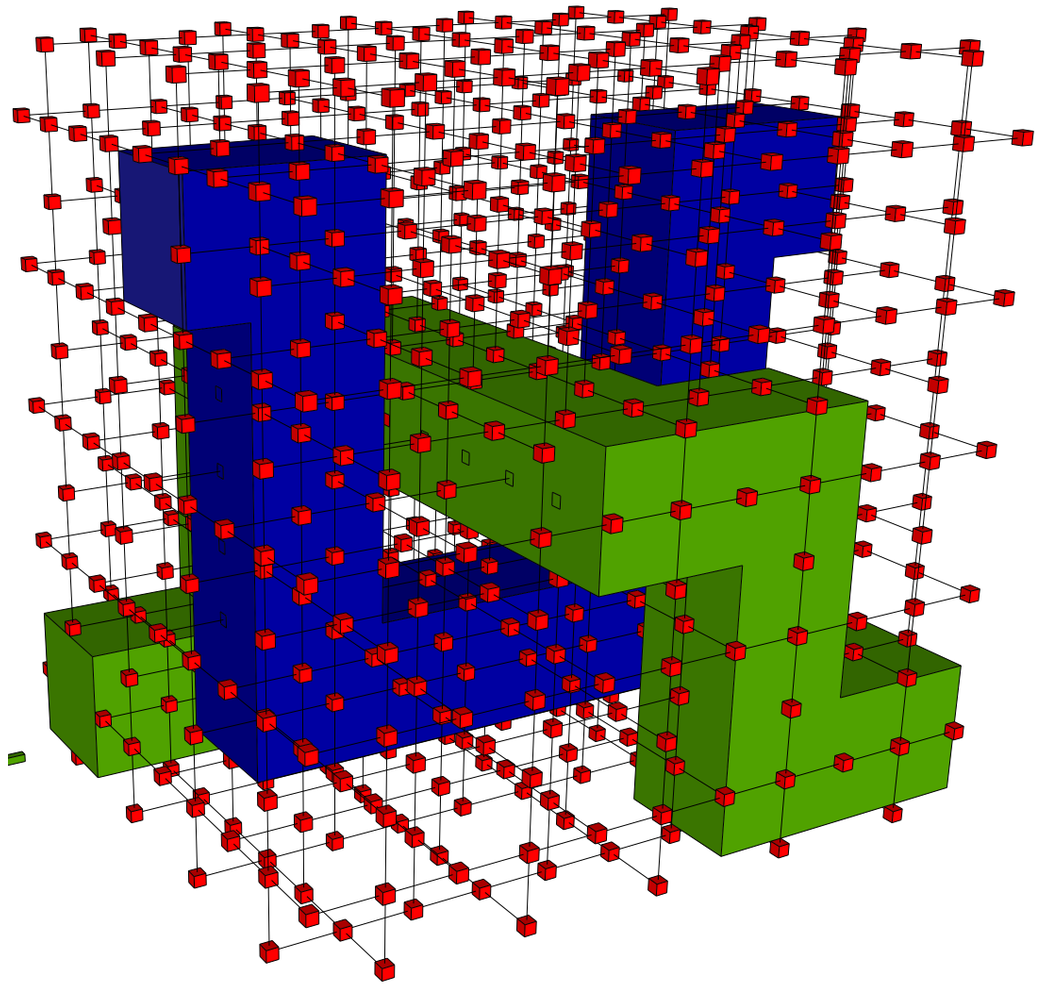
- Expected number of runs to find a solution: $\#\mathcal{O} = \pi 2^{14} \sqrt{10 \cdot D}$

# Quantum attacks on mining

- Enter physics
  - Gates take time
  - Gates are noisy—-> need error correction
  - Assume Raussendorf foliation of toric code (this is what IBM, Google, Psiquantum & others plan to use)




Raussendorf, Harrington, Goyal, Annals Phys. (2016)



Paler et al. Sci. Rep. 4, 4657 (2014)

# Quantum attacks on mining

- Classical miners are defined by Hashing rate (number Hashes/second)
  - Algorithm Specific Integrated Circuits (ASICs) are fast!


$h_{CL}$= 14TH/s

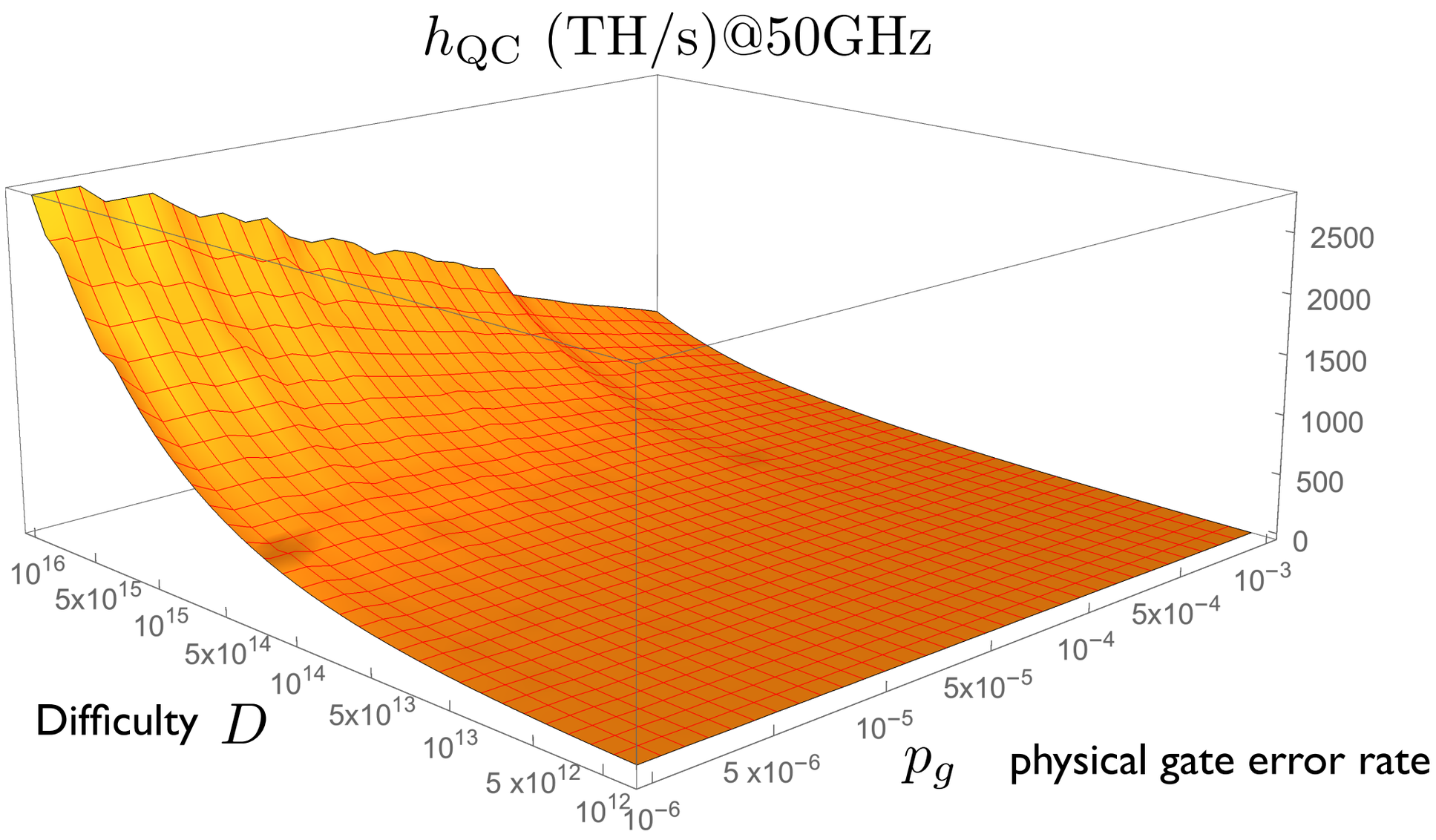
- Quantum Effective Hashing rate: Expected number of Hashes on a classical computer divided by expected time to find a solution on quantum computer

$$h_{\mathrm{QC}} \equiv \frac{N/t}{\tau} = \frac{0.28 \times s\sqrt{D}}{c_\tau(D, p_g)}$$

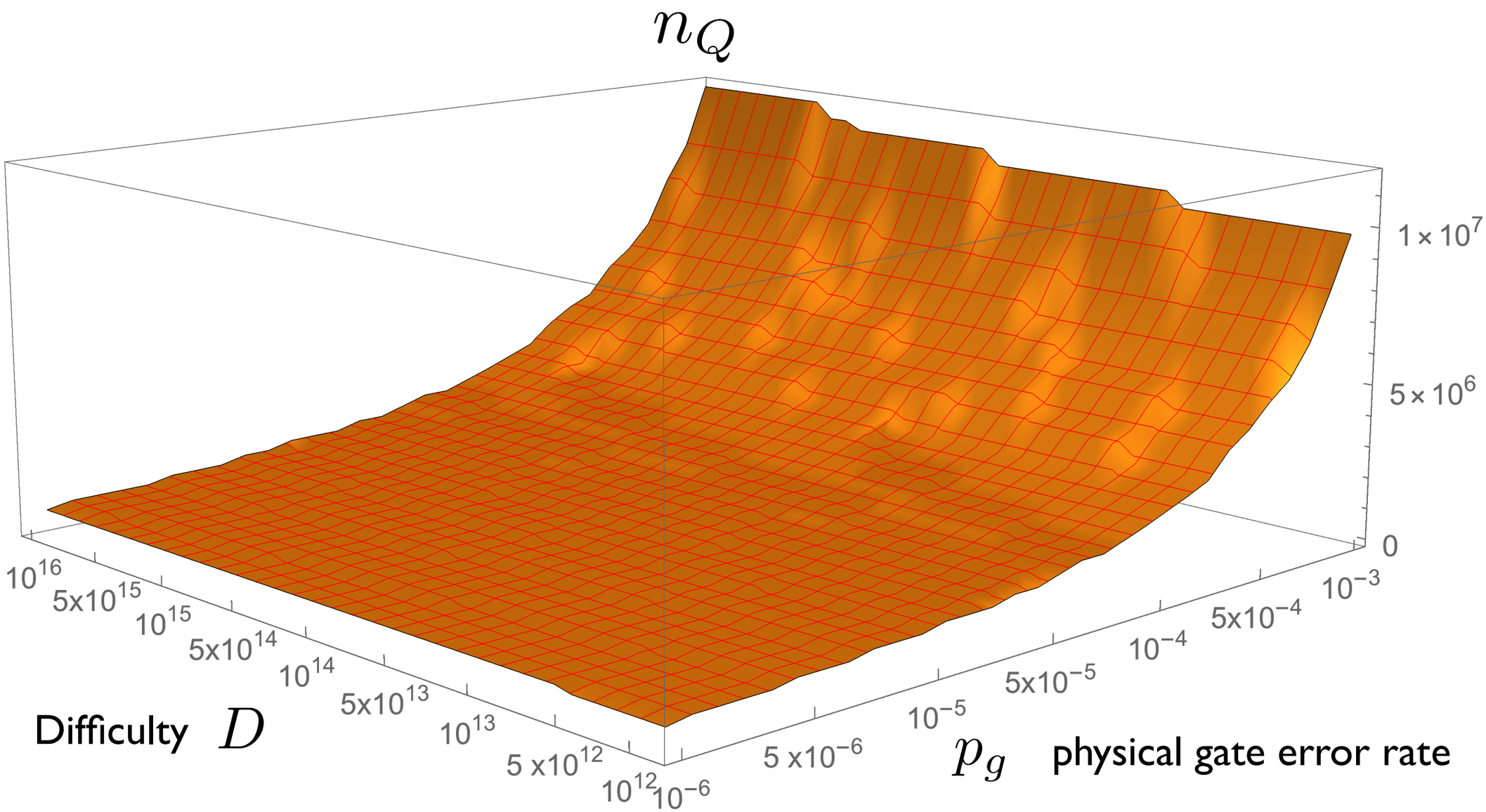
s: clock speed

Overhead depending on difficulty and gate error rate

# Quantum effective Hashing rate



# Number of physical qubits needed



# Quantum attacks on mining

- Quantum computers are slow…..
  - Assuming clock speed of $s = 66.7\text{MHz}$ , gate error rate $p_g = 5 \times 10^{-4}$
  - Effective Hash rate

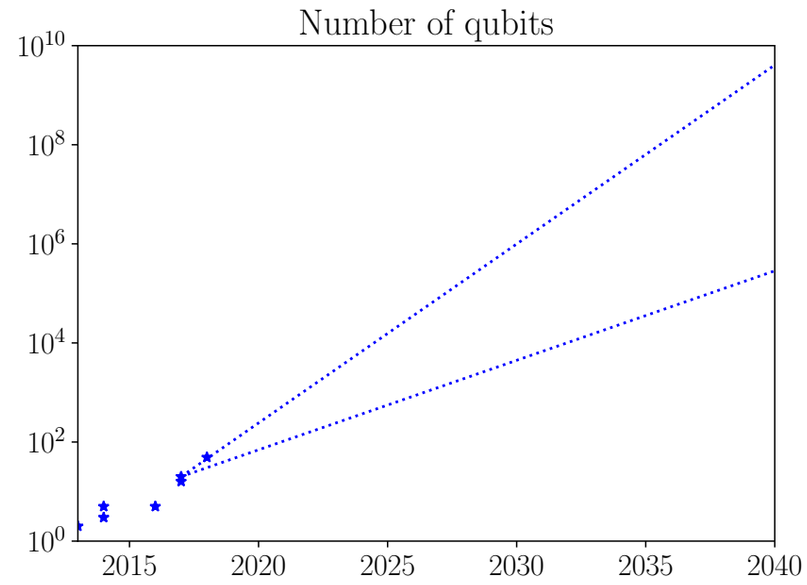
$$h_{\text{QC}} = 13.8\text{GH/s}$$

  - using over 4.4 million qubits
  - 1000 times slower than current ASICs!

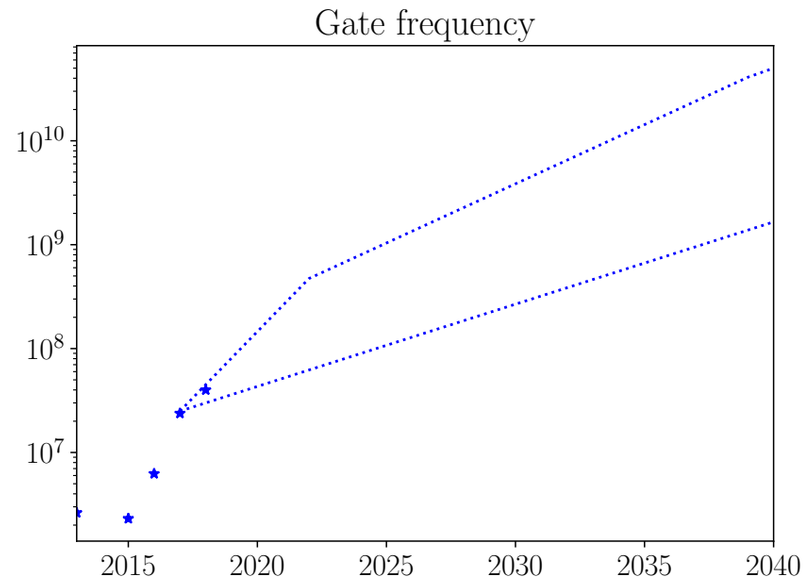
- Parallelization doesn't help much
  - Using *d* quantum computers in parallel effective Hash rate improves by a factor $\sqrt{d}$.

# Quantum Moore's Law

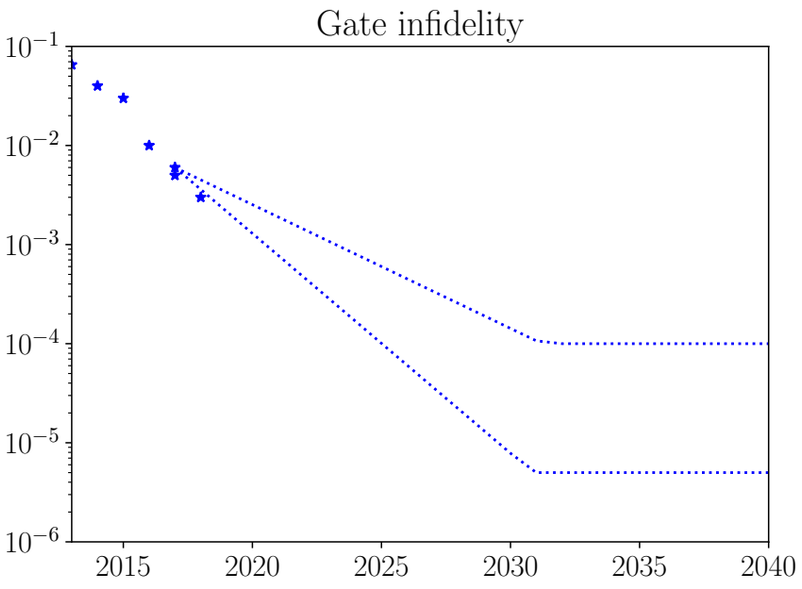
- Prospectus on improvements in gates speed, gate fidelity, number of qubits
- Model optimistic and pessimistic improvements



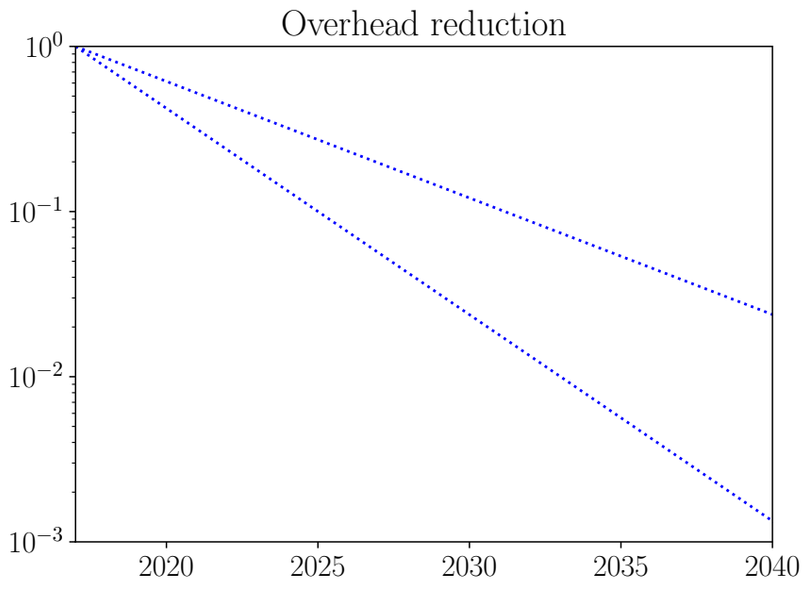

(a)




(b)



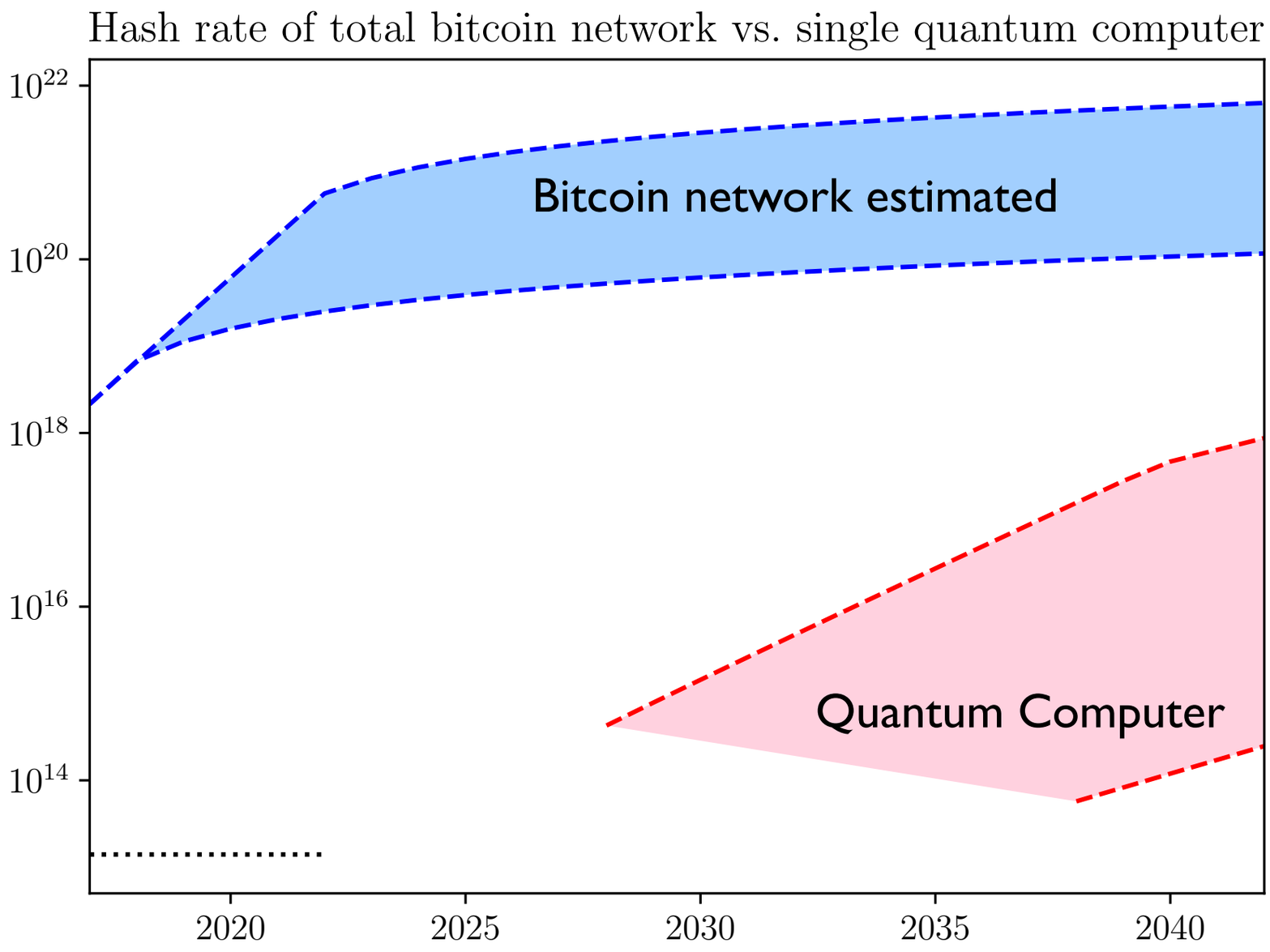

(c)




(d)

# Quantum attacks on mining

- Prospectus based on a quantum Moores' law and classical Moores' law




- One or many quantum computers will not dominate the network

# Countermeasures for mining attacks

- Consider a scenario where countries ban power intensive classical mining operations so that quantum computers could have competitive advantage
- Reduce relative advantage of quantum computers by adding structure to the mining problem
- *Momentum* proof of work
  - Two conditions must be met
  - Like before, find a header *H* for a Hash function $h_1 : \{0,1\}^* \to \{0,1\}^n$ such that $h_1(H \parallel x) \leq t$
  - Header must also find collision for another Hash function $h_2 : \{0,1\}^* \to \{0,1\}^\ell$

$$\lfloor h_2(H \parallel a) = h_2(H \parallel b) \text{ and } a, b \leq 2^\ell \qquad n \leq \ell$$

# Countermeasures for mining attacks

- Problem: Find Header triple (*H||a||b)* such that

$$h_1(H \parallel a \parallel b) \leq t \text{ and } h_2(H \parallel a) = h_2(H \parallel b) \text{ and } a, b \leq 2^{\ell} \qquad n \leq \ell$$

- Classical Algorithm
  - Pick subset $S \subset \{0,1\}^{\ell}$ and evaluate $h_2(H \parallel a)$ for all $a \in S$
  - Expected number of collisions pairs (a,b): $\binom{|S|}{2} \times \frac{2^l}{2^{2l}} \approx \frac{|S|^2}{2^l}$
  - All collisions found in time $|S|$
- For each collision check if $h_1(H \parallel a \parallel b) \leq t$
  - Would need to check on average $2^n/t$ triples (*H||a||b)*
  - Total number of Headers *H* to try is then $m = \max\{1, \frac{2^{n+\ell}}{t|S|^2}\}$
- Run time is $m|S|$, minimized when $m = 1$, $|S| = \sqrt{2^{n+\ell}/t}$
- Total run time $T = \sqrt{2^{n+\ell}/t}$

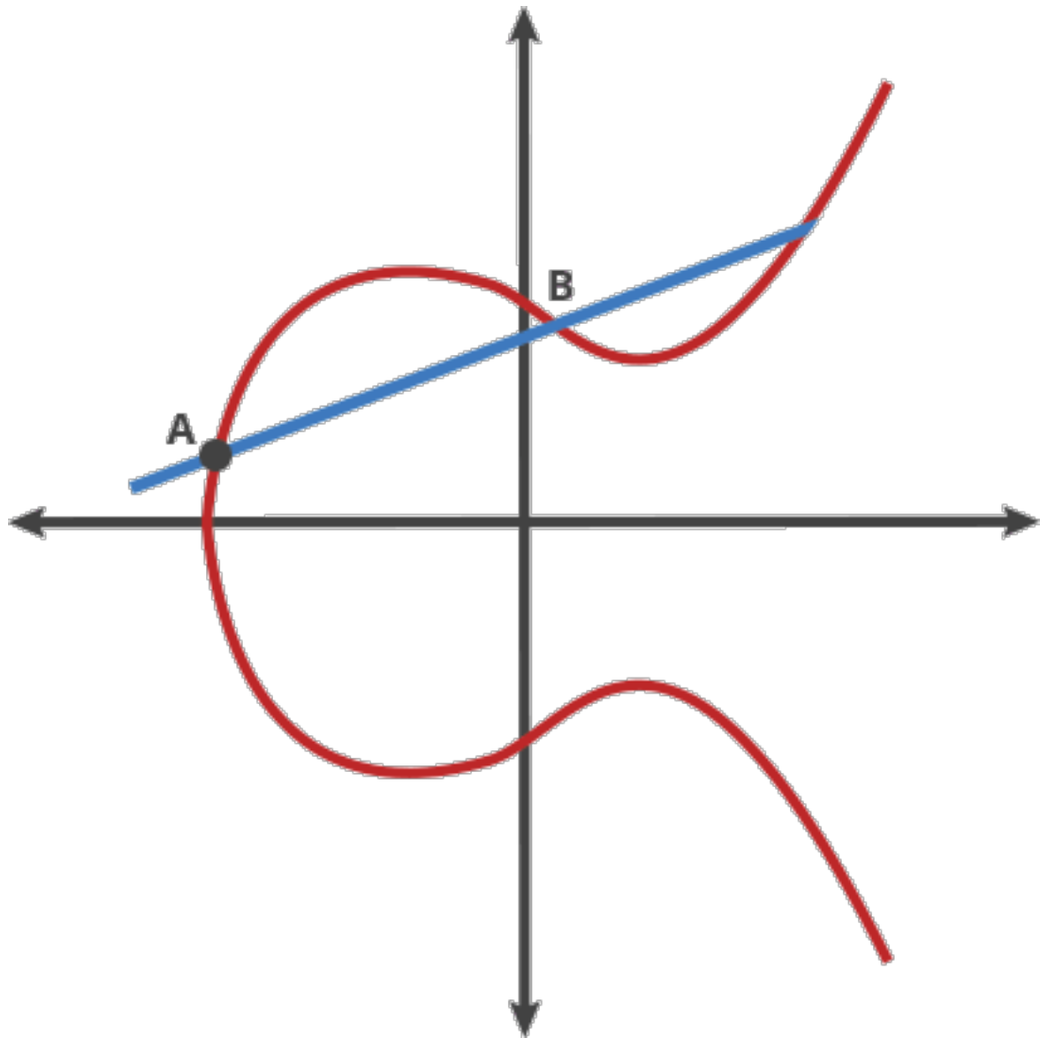
# Countermeasures for mining attacks

- Quantum Algorithm
  - Time to find a collision is lower bounded* by $|S|^{2/3}$; this bound is tight using element distinctness algorithm**
  - Number of headers *H* to try is still $m = \max\{1, \frac{2^{n+\ell}}{t|S|^2}\}$
  - By Grover search, run time to find good header *H* is $O(\sqrt{m})$
  - Total run time is then $O(\sqrt{m}|S|^{2/3})$
  - Run time minimised again when $m = 1$ $|S| = \sqrt{2^{n+\ell}/t}$
  - Total run time is $T^{2/3}$, much longer than $T^{1/2}$ without collisions

*Aaronson and Shi (2004)
**Ambainis (2007)

# Quantum attacks on signatures

- Bitcoin and many cryptocurrencies use Elliptic curve cryptography for digital signatures based on the hardness of solving the Elliptic Curve Discrete Log Problem
- Use n=256 bit digital signatures. Best known classical algorithms are exponential time in n.




Discrete Log problem:
Given A,Z *solve for x in*
$A^x=Z$

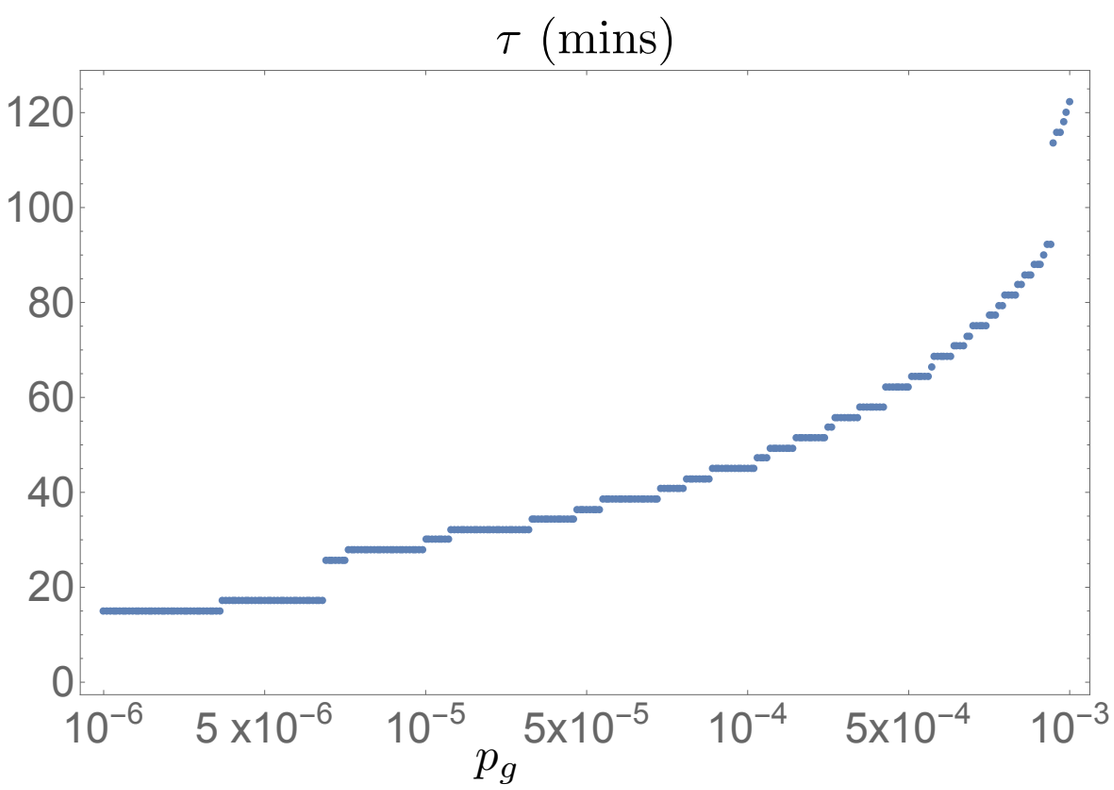
# Quantum attacks on signatures

- Shor found a polynomial time quantum algorithm to solve Discrete Log Problem in $O(n^3)$ gates.
- Biggest threat:
  - After a transaction has been broadcast to the network but before it is placed on the blockchain it is at risk from quantum attack.
  - If secret key can be derived from broadcast public key before transaction is placed on the blockchain then an attacker could use this secret key to broadcast a new transaction from same address to her own.
  - Attacker ensures new transaction is placed on blockchain first by offering high transaction fee. Unstoppable theft!

# Quantum attacks on signatures

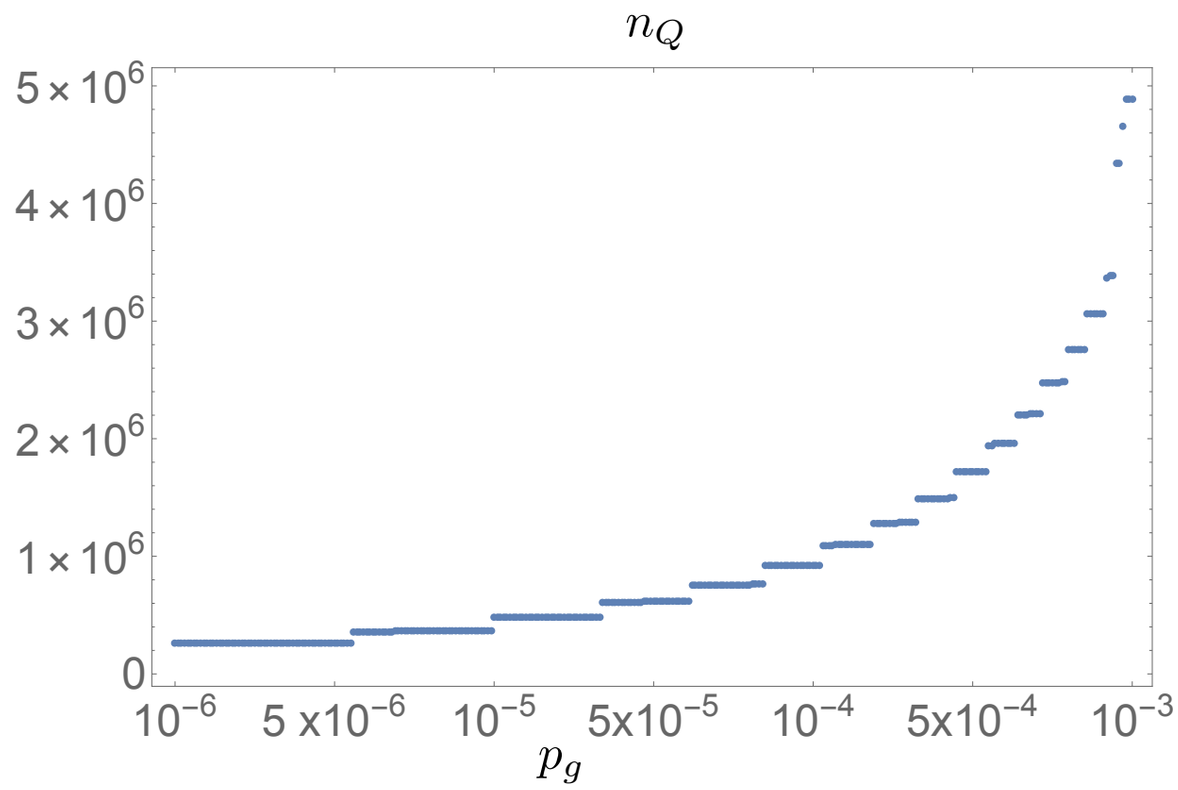
- 10GHz clock speed quantum computer attacking signatures. If cracked in time of ~10-20 minutes then theft possible

cracking time




physical gate error rate

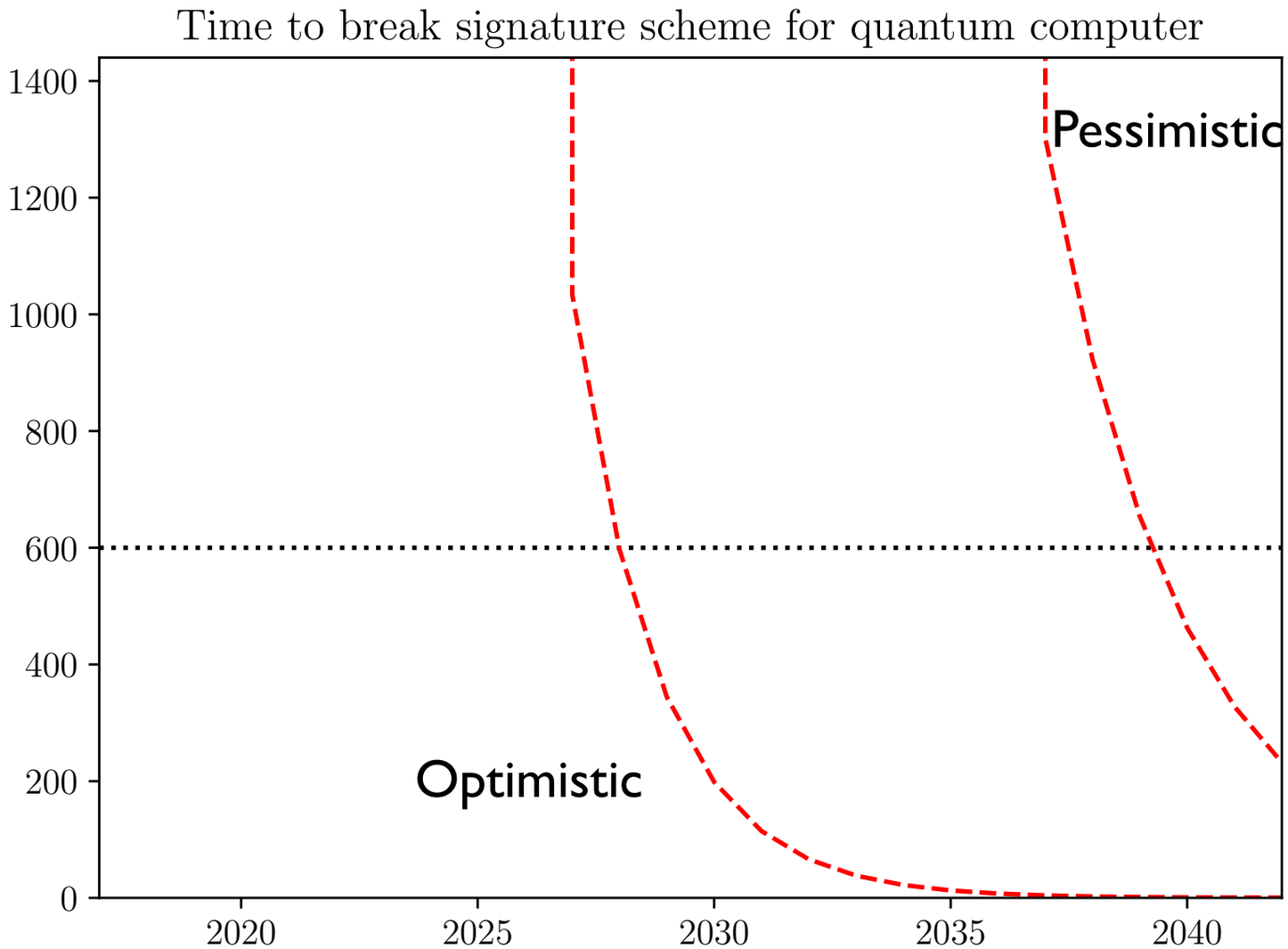
number of physical qubits




physical gate error rate

# Quantum attacks on signatures

- Prospectus to crack digital signatures




- Under quite optimistic scenario could be cracked in 10 years

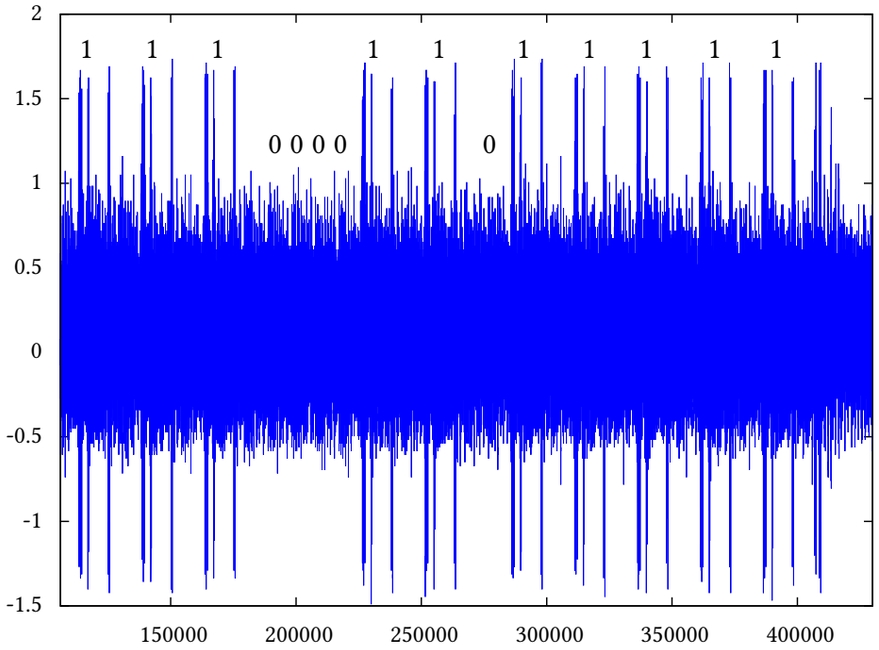
# Countermeasures for signature attacks

- Post-quantum crypto
  - Many schemes have less structure than elliptic curve problem
  - Harder for quantum computers but also slower transactions

| type | name | classical security (bits) | quantum security (bits) | PK length (kb) | signature length (kb) | total length (kb) |
|---|---|---|---|---|---|---|
| | ECDSA | 127 | 0 | 0.3 | 0.5 | 0.8 |
| I.1 | GPV[45] | 100 | | 300 | 240 | 540 |
| I.2 | LYU[45] | 100 | | 65 | 103 | 168 |
| I.3 | BLISS[37] | 128 | | 7 | 5 | 12 |
| I.4 | FALCON-512*[46] | 114 | 103 | 7.2 | 4.9 | 12.1 |
| I.5 | ring-TESLA[38] | 128 | | 26.6 | 11.9 | 38.5 |
| I.6 | qTESLA-128*[47] | 128 | | 23.8 | 21.7 | 45.4 |
| I.7 | DILITHIUM*[39] | 138 | 125 | 11.8 | 21.6 | 33.4 |
| II.1 | RAINBOW[48] | 160 | | 305 | 0.2 | 305.2 |
| III.1 | LMS[49] | 256 | 128 | 0.8 | 22.6 | 23.4 |
| III.2 | XMSS[29] | 196 | 93 | 13.6 | 22.3 | 35.9 |
| III.3 | SPHINCS[30] | 256 | 128 | 8.4 | 328 | 336.4 |
| III.4 | NSW[31] | 128 | | 0.3 | 36 | 36.3 |
| IV.1 | CFS[32] | 83 | | 9216 | 0.1 | 9216.1 |
| IV.2 | QUARTZ[33] | 80 | | 568 | 0.1 | 568.1 |

# Post-Quantum Vulnerabilities

- Example: Side channels attack on a lattice based cryptography BLISS
  - In sampling steps of protocol, the relative norm of the secret key is leaked
  - Modern CPUs include branch tracing which count CPU cycles, context switches, etc. What is meant to look random is not computationally!
  - Can also use electromagnetic tracing: H-field probe at 30MHz-3GHz



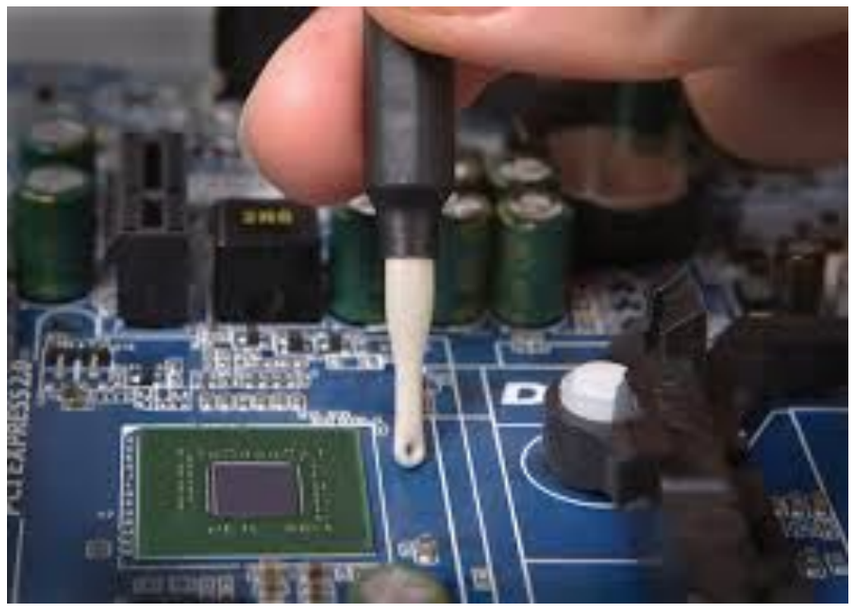



**Figure 6: Electromagnetic measure of BLISS rejection sampling for norm 14404.**

- Physics vs Computer Science vs Physics vs ….;