QUANTUM ATTACKS ON AES

7/9/24

When do we need to worry about a structureless, quantum, known plaintext attack against AES?

Samuel Jaques

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Rocco Ceselin/Google

MAIN QUESTION

When do we need to worry about a structureless, quantum, known plaintext attack against AES?

Attacking Block Ciphers

Known plaintext attack: Given $O(1)$ pairs of m_i and $c_i = E_k(m_i)$ for a fixed key k, recover k

- Not the only symmetric key attack!
  - Multi-target attacks: (many such pairs, any key is fine)
  - Unknown plaintext (we must guess m_i as well)
  - Leakage attacks (we learned some aspect of internal state)
  - Fault attacks, etc.
- Nearly identical cost as hash pre-image attacks

Structureless Attacks

- I assume we use none of the internal structure. This excludes:
  - Differential cryptanalysis
  - Linear cryptanalysis
  - Period-finding attacks on (e.g.) Evan-Mansour Constructions
- Quantum analogues of these techniques exist:
  - Kuwakado and Morii. Security on the quantum-type Even-Mansour cipher, in ISITA 2012.
  - Kaplan, Leurent, Leverrier, Naya-Plasencia. Breaking Symmetric Cryptosystems Using Quantum Period Finding, in Crypto 2016.
  - Kaplan, Leurent, Leverrier, Naya-Plasencia. Quantum Differential and Linear Cryptanalysis, in TSC 2016.
  - (And many more!)

Classical Structureless Attack

Just guess and check:

```
For $k' = 0$ to $k' = 2^n - 1$:
    If $E_{k'}(m_i) = c_i$ for all $(m_i, c_i)$, return $k'$
```

Expected running time: $O(2^n)$

Exponential, therefore secure*

*to be revisited!

Quantum Structureless Attack: Grover

Grover’s algorithm:

```
For $i = 0$ to $k' = \sqrt{2^n}$:
    Apply a “diffusion operator” // cheap quantum magic
    Apply $E_{(*)}(m_i)$ in superposition and check the result
Measure the output $k'$
Return $k'$
```

Expected runtime: $O(\sqrt{2^n}) = O(2^{n/2})$.

Square root speed-up!

How much of a threat is this?

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

Quantum attack is exponential, $O(2^{n/2})$, so...

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

Quantum attack is exponential, $O(2^{n/2})$, so…

- $n = 1$ is safe today

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

Quantum attack is exponential, $O(2^{n/2})$, so...

- $n = 1$ is safe today
- $n = 64$ is about as safe as RSA

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

Quantum attack is exponential, $O(2^{n/2})$, so…

- $n = 1$ is safe today
- $n = 64$ is about as safe as RSA
- $n = 128$ gives a 2^{64} attack… is that safe?

How much of a threat is this?

The classical attack is exponential, $O(2^n)$, but:

- If $n = 56$ (i.e., DES) that's way too easy
- If $n = 64$, that's *probably* too easy
- If $n \geq 128$ this seems to be safe

Quantum attack is exponential, $O(2^{n/2})$, so…

- $n = 1$ is safe today
- $n = 64$ is about as safe as RSA
- $n = 128$ gives a 2^{64} attack… is that safe?
  - …is it really 2^{64} or a higher constant?

Grover’s Algorithm Constants

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover’s Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

Grover’s Algorithm Constants

- To decide on the actual cost, we need the constants of the $O(2^{n/2})$ runtime

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover’s Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

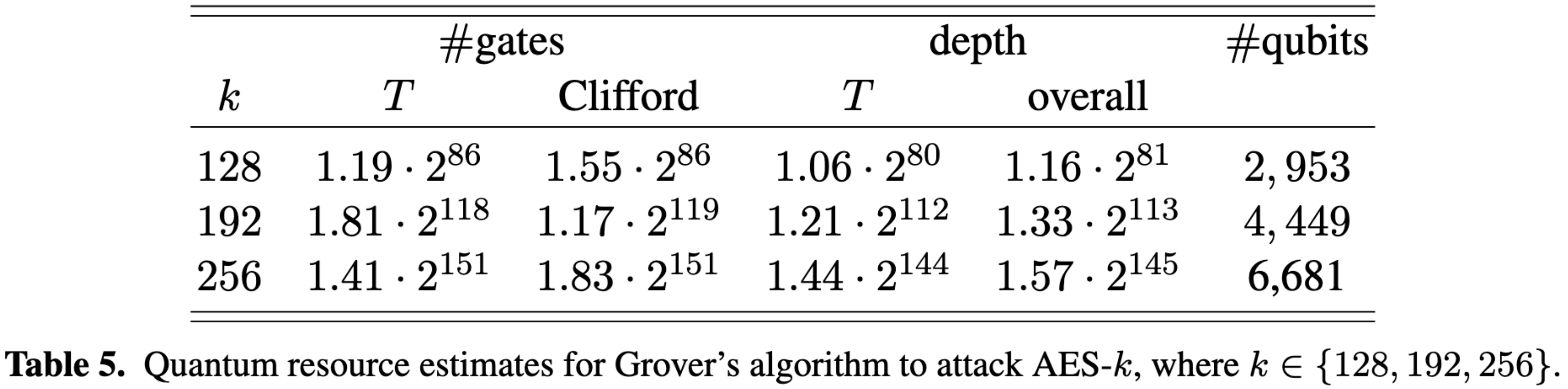
Grover's Algorithm Constants

- To decide on the actual cost, we need the constants of the $O(2^{n/2})$ runtime
- To find those, we would need to design a quantum circuit for AES

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover's Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

Grover’s Algorithm Constants

- To decide on the actual cost, we need the constants of the $O(2^{n/2})$ runtime
- To find those, we would need to design a quantum circuit for AES
- Luckily, people have! So we check this from 2015:



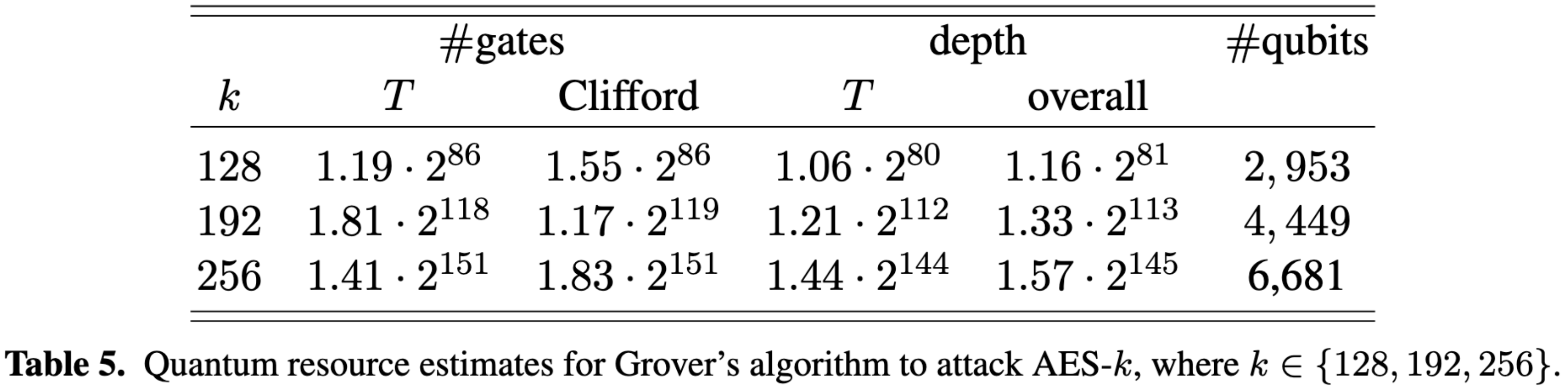
	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	$2,953$
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	$4,449$
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Table 5. Quantum resource estimates for Grover’s algorithm to attack AES-k, where $k \in \{128, 192, 256\}$.

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover’s Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

Grover’s Algorithm Constants

- To decide on the actual cost, we need the constants of the $O(2^{n/2})$ runtime
- To find those, we would need to design a quantum circuit for AES
- Luckily, people have! So we check this from 2015:



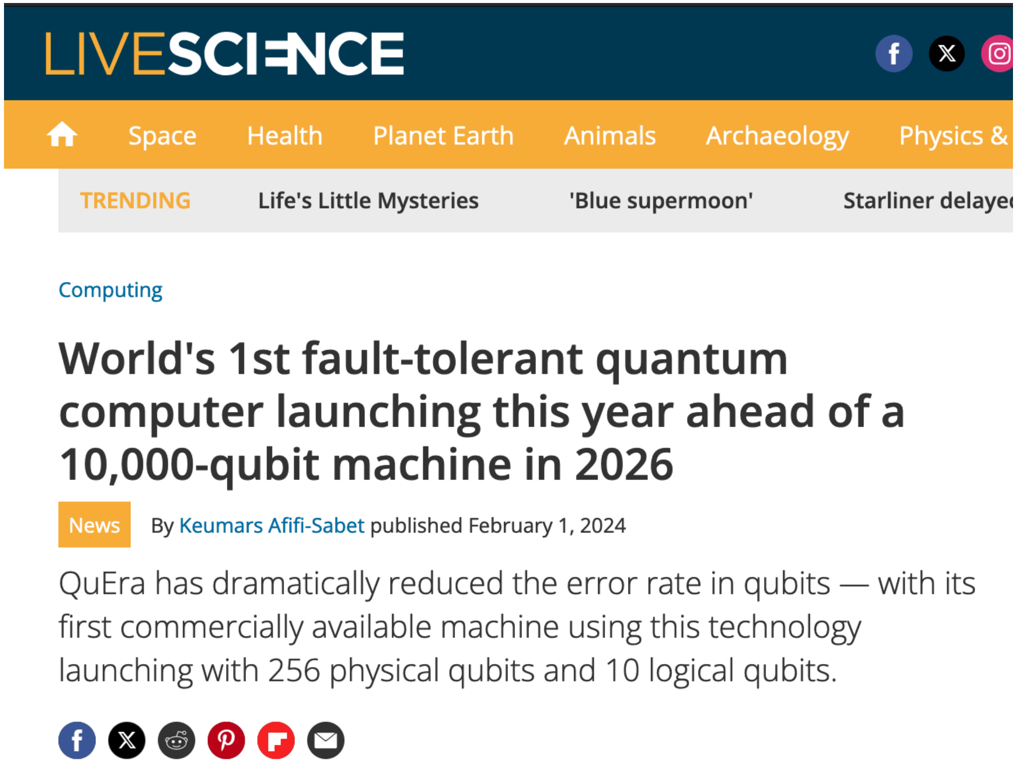
	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2, 953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4, 449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Table 5. Quantum resource estimates for Grover’s algorithm to attack AES-k, where $k \in \{128, 192, 256\}$.

Only 2,953 qubits!?

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover’s Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

Quantum Computing News



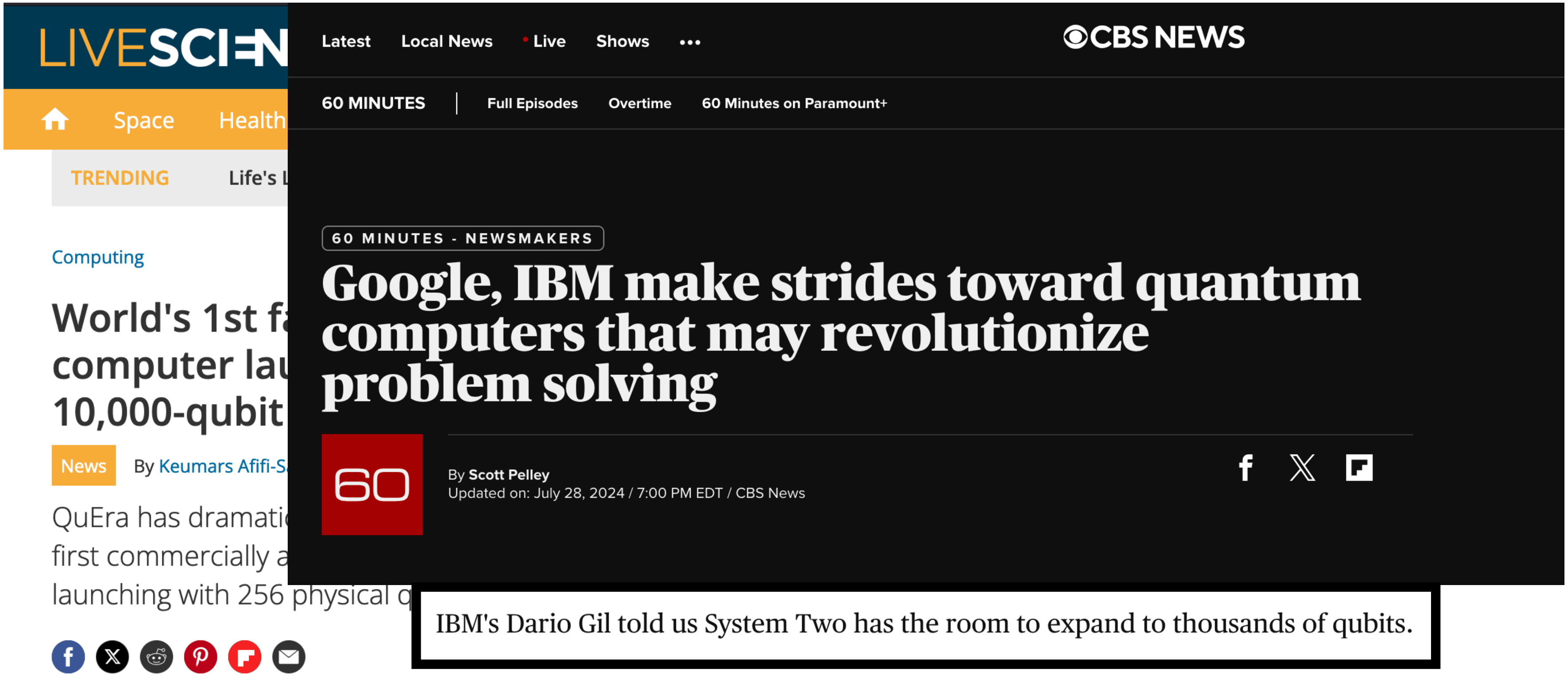
Computing

World's 1st fault-tolerant quantum computer launching this year ahead of a 10,000-qubit machine in 2026

News By Keumars Afifi-Sabet published February 1, 2024

QuEra has dramatically reduced the error rate in qubits — with its first commercially available machine using this technology launching with 256 physical qubits and 10 logical qubits.

Quantum Computing News



Computing

World's 1st f
computer lau
10,000-qubit

News By Keumars Afifi-S

QuEra has dramati
first commercially a
launching with 256 physical q

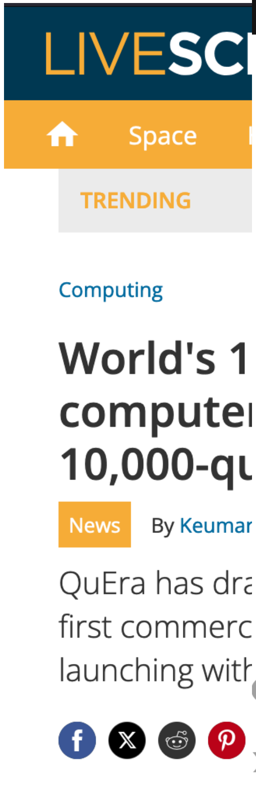
60 MINUTES - NEWSMAKERS

Google, IBM make strides toward quantum computers that may revolutionize problem solving

By **Scott Pelley**
Updated on: July 28, 2024 / 7:00 PM EDT / CBS News

IBM's Dario Gil told us System Two has the room to expand to thousands of qubits.

Quantum Computing News

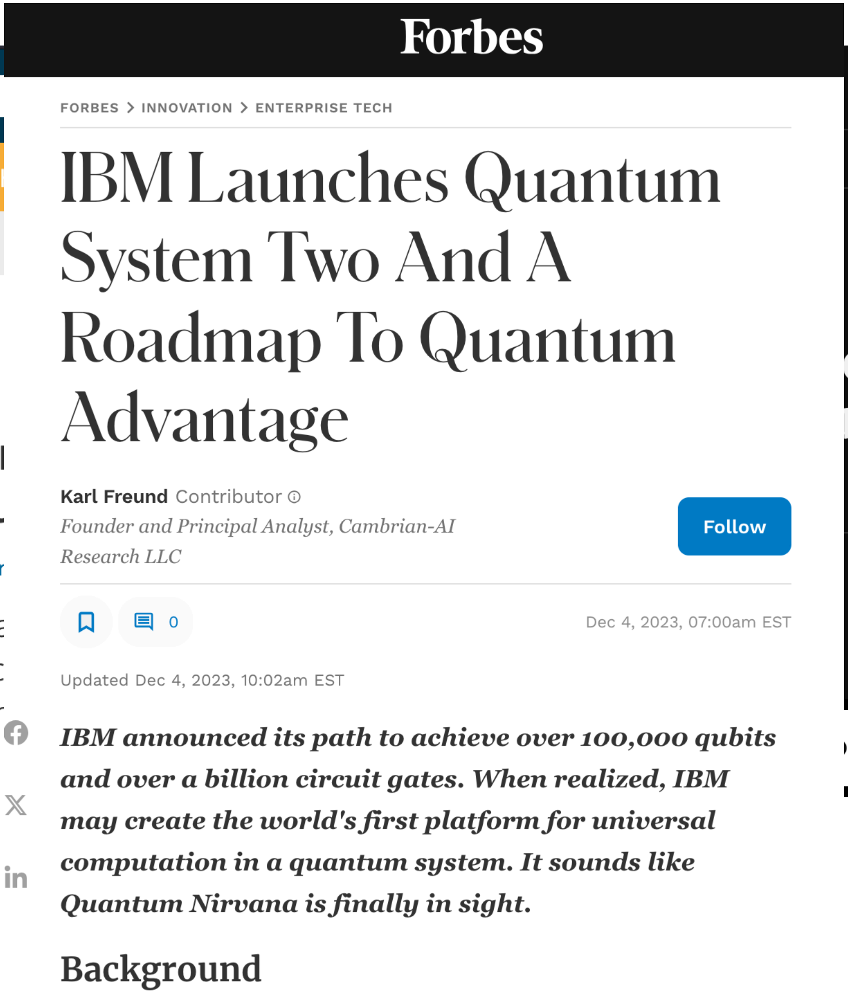


Computing

World's 1
compute
10,000-qu

News By Keumar

QuEra has dra
first commerc
launching with



FORBES > INNOVATION > ENTERPRISE TECH

IBM Launches Quantum System Two And A Roadmap To Quantum Advantage

Karl Freund Contributor

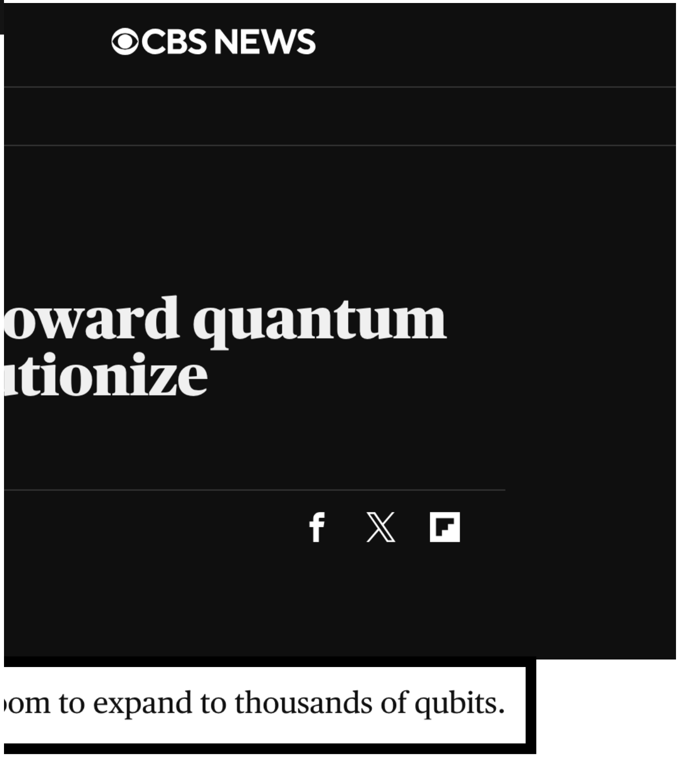
Founder and Principal Analyst, Cambrian-AI Research LLC

Dec 4, 2023, 07:00am EST

Updated Dec 4, 2023, 10:02am EST

IBM announced its path to achieve over 100,000 qubits and over a billion circuit gates. When realized, IBM may create the world's first platform for universal computation in a quantum system. It sounds like Quantum Nirvana is finally in sight.

Background



oward quantum
tionize

om to expand to thousands of qubits.

Computation Time

- 2^{86} gates: is that a lot?
- The bitcoin network does 2^{69} hashes per second
- The bitcoin network can compute 2^{86} hashes in 36 hours

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Table 5. Quantum resource estimates for Grover's algorithm to attack AES-k, where $k \in \{128, 192, 256\}$

BTC Hashrate: 596.04 EH/s

Aug 21, 2024 07:59 PM UTC - 596,039,618,890,653,500,000 H/s

Grassl, Langenberg, Roetteler, and Steinwandt. Applying Grover's Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016

A reasonable conclusion someone could make from all this:

“Grover’s algorithm can break AES-128 roughly at the scale of ‘next year’s quantum computers’ and ‘the bitcoin network’. Maybe we need to move away from 128 bit keys right away?

A reasonable conclusion someone could make from all this:

"Grover's algorithm can break AES-128 roughly at the scale of 'next year's quantum computers' and 'the bitcoin network'. Maybe we need to move away from 128 bit keys right away?

This is completely incorrect

A reasonable conclusion someone could make from all this:

“Grover’s algorithm can break AES-128 roughly at the scale of ‘next year’s quantum computers’ and ‘the bitcoin network’. Maybe we need to move away from 128 bit keys right away?

This is completely incorrect

My own opinion:

A reasonable conclusion someone could make from all this:

“Grover’s algorithm can break AES-128 roughly at the scale of ‘next year’s quantum computers’ and ‘the bitcoin network’. Maybe we need to move away from 128 bit keys right away?

This is completely incorrect

My own opinion:

“Grover’s algorithm will not break AES-128 in our lifetimes, and will probably never break it.”

A reasonable conclusion someone could make from all this:

“Grover’s algorithm can break AES-128 roughly at the scale of ‘next year’s quantum computers’ and ‘the bitcoin network’. Maybe we need to move away from 128 bit keys right away?

This is completely incorrect

My own opinion:

“Grover’s algorithm will not break AES-128 in our lifetimes, and will probably never break it.”

This talk: walking through everything wrong with the first conclusion

MISCONCEPTION
#1

Misconception: Qubits are the limiting factor for quantum circuits

QUANTUM COMPUTERS

A quick introduction

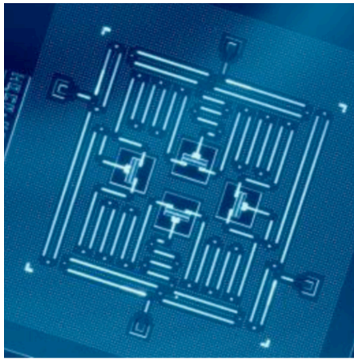
Basics: Qubits

A **qubit** is a device that holds **quantum data**, which can be $|0\rangle$, $|1\rangle$, or any complex linear combination of the two (normalized to 1), e.g. $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, or $\frac{1}{2}|0\rangle - i\frac{\sqrt{3}}{2}|1\rangle$

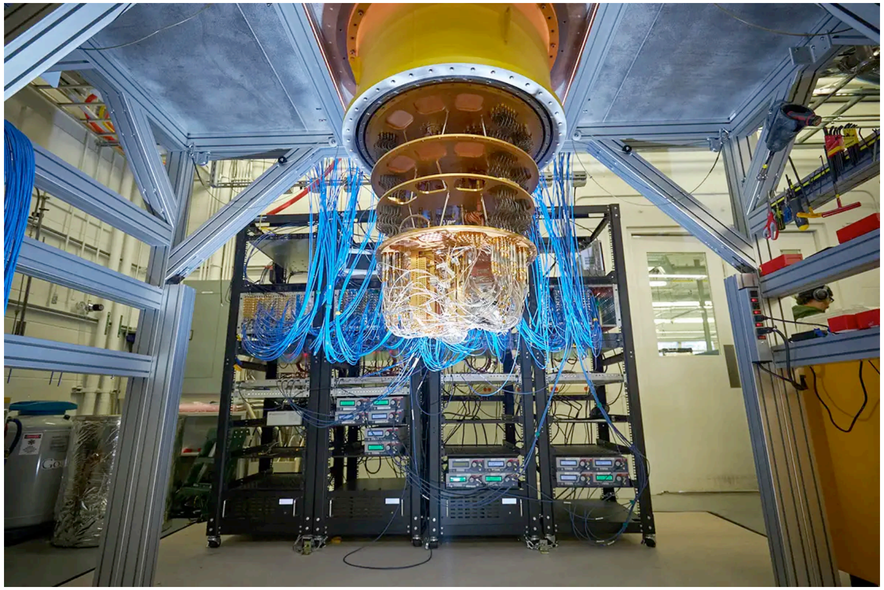
Qubit Types

Any “two-level” quantum system can be a qubit:

Superconducting qubits: A superconducting wire with current flowing in one direction or another



Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen, 2017

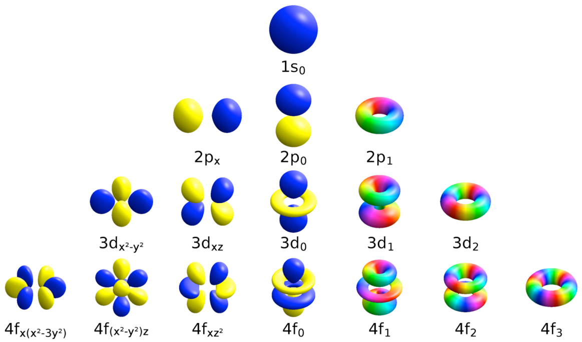


Rocco Ceselin/Google

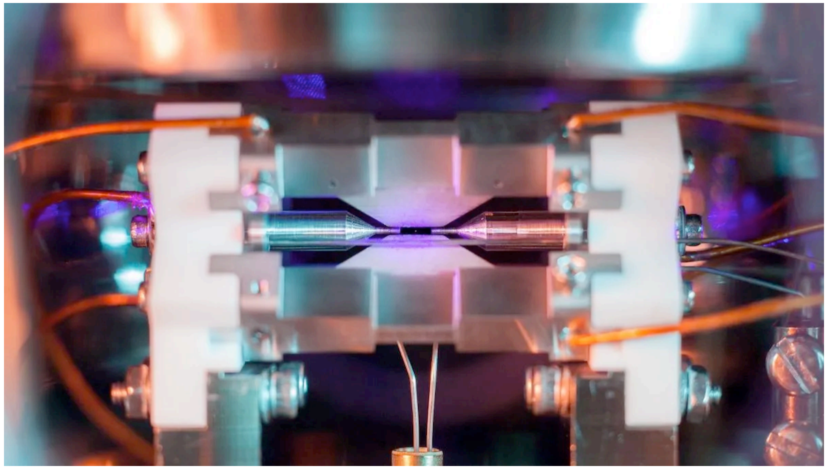
Qubit Types

Any “two-level” quantum system can be a qubit:

Trapped ion qubits: an atom where electrons are either in a high or low energy orbital



Wikipedia user Geek3

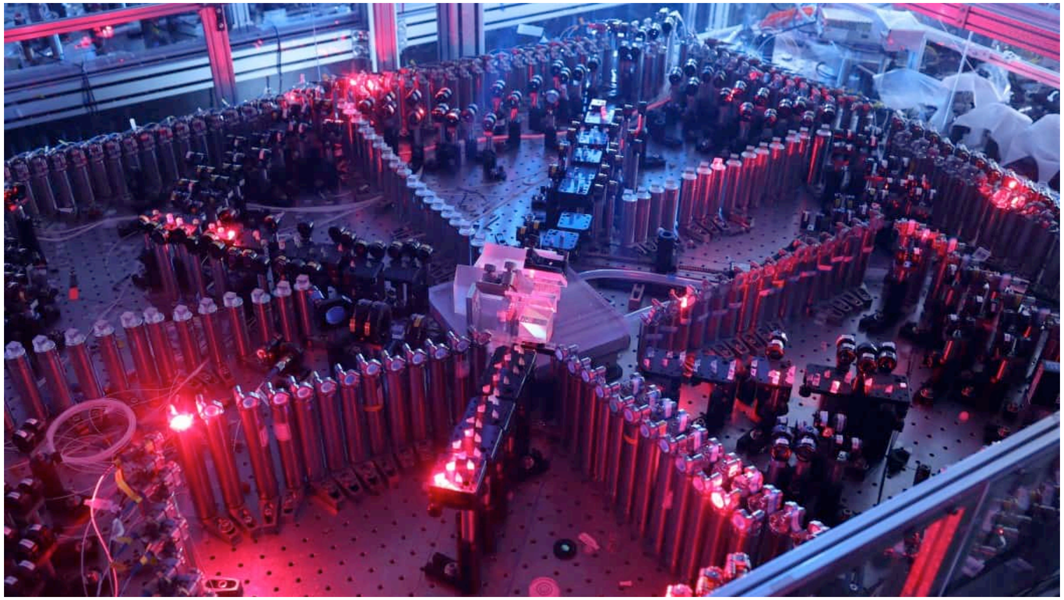


David Nadlinger

Qubit Types

Any "two-level" quantum system can be a qubit:

Photonic qubits: a photon that could be in one of two physical locations (e.g. fibre optic cables)

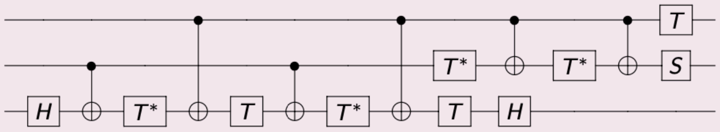


Chao-Yung Lu

Basics: Gates

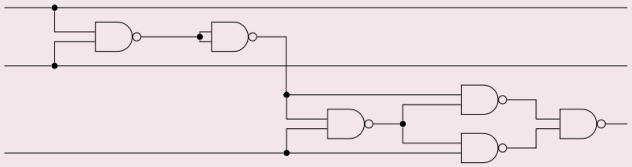
We manipulate the qubits with **gates**, which change the quantum data. Analogous to classical gates, but they are almost always a **process**, not a **device**.

Quantum



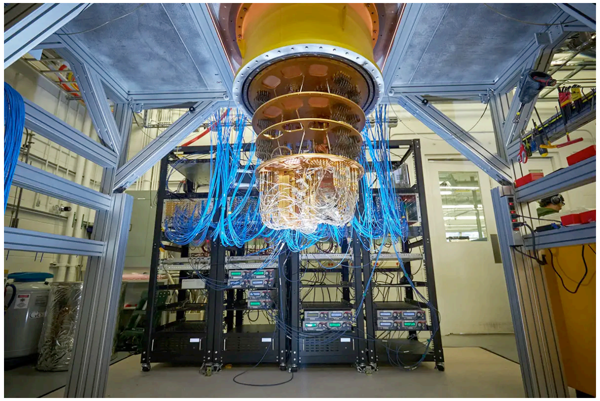
Time→

Classical



Time/Space→

Basics: Noise

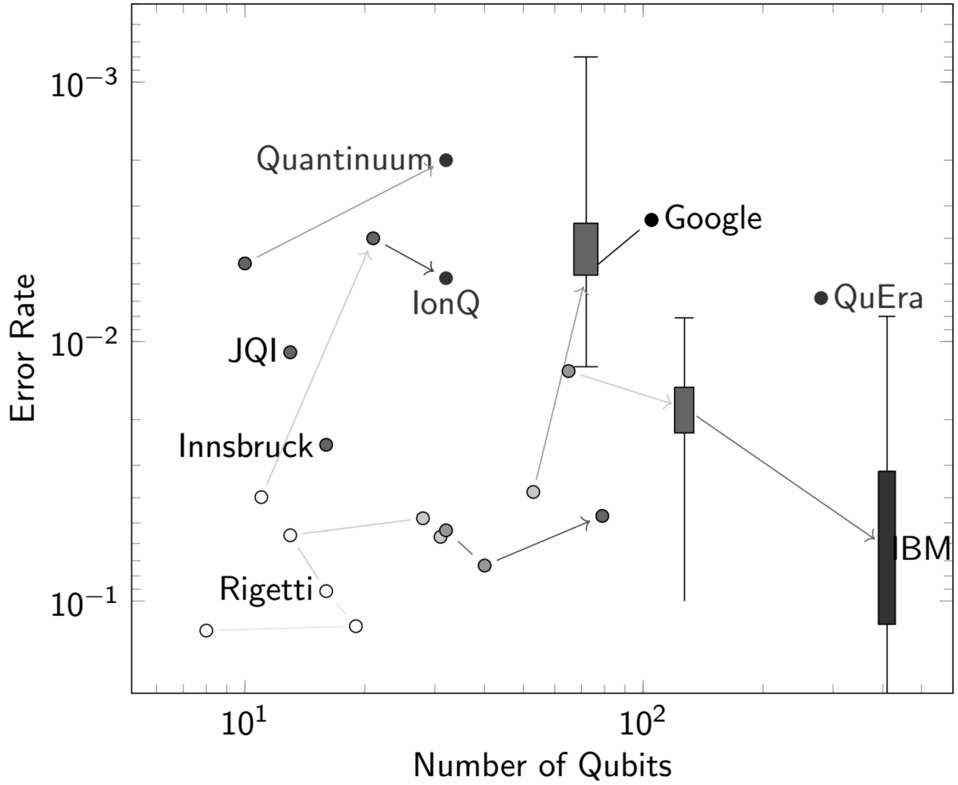


Rocco Ceselin/Google

Qubits are highly susceptible to noise. Noise is any uncontrolled process which modifies the quantum data.

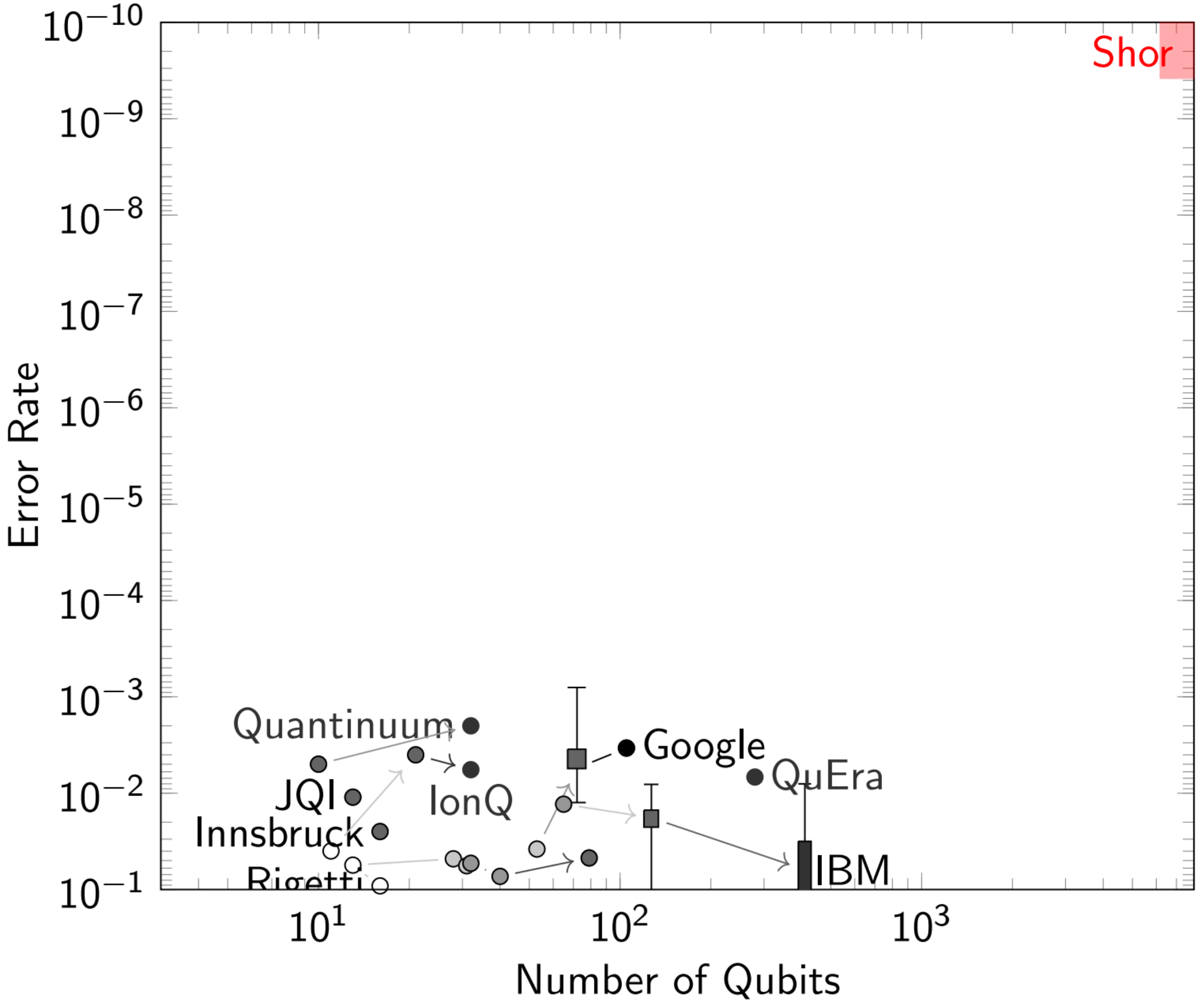
- Classical noise is much easier to deal with: absorbing a small bit of energy won't flip a bit. For qubits, any unwanted interaction causes problems
- Qubits can have "bit flip errors" (similar to classical bit flip) but also "phase flip errors" (no classical analogue) or **any linear combination of the two types**

Quantum Computing Today

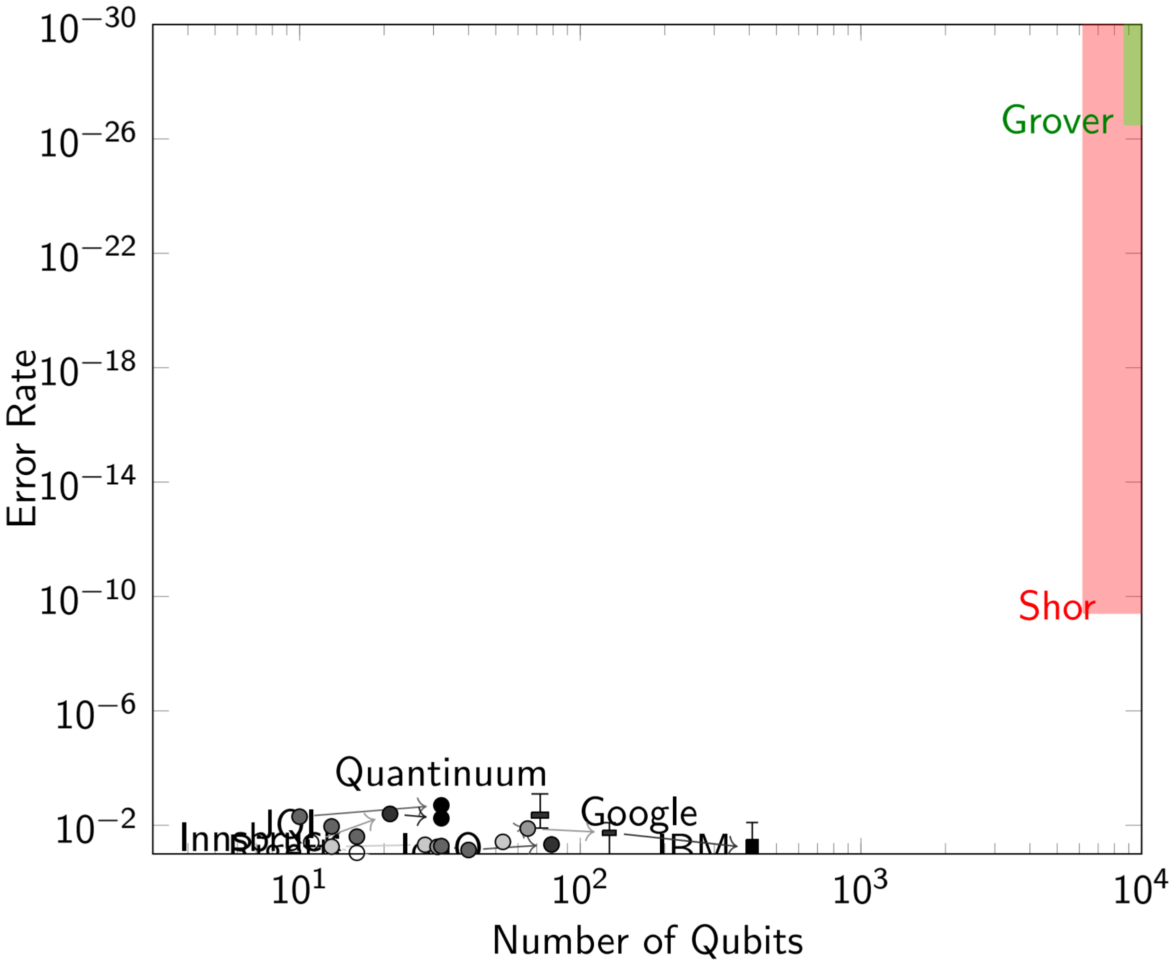


(I had to make dubious assumptions to compress "error rate" to a single number; this is not super precise)

Quantum Computing Today



Quantum Computing Today



Error Correcting Codes

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
    - **Physical qubits**: physical devices like today’s qubits

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
  - **Physical qubits**: physical devices like today's qubits
  - **Logical qubits**: an abstraction representing the collection of qubits in a code that act like one high-fidelity qubit

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
  - **Physical qubits**: physical devices like today's qubits
  - **Logical qubits**: an abstraction representing the collection of qubits in a code that act like one high-fidelity qubit

Basic assumption:

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
  - **Physical qubits**: physical devices like today's qubits
  - **Logical qubits**: an abstraction representing the collection of qubits in a code that act like one high-fidelity qubit

Basic assumption:

1 qubit with error rates a **billion** times better than today

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
  - **Physical qubits**: physical devices like today's qubits
  - **Logical qubits**: an abstraction representing the collection of qubits in a code that act like one high-fidelity qubit

Basic assumption:

1 qubit with error rates a **billion** times better than today

Is much harder than

Error Correcting Codes

- Quantum error correcting codes are like classical error correcting codes: we protect against noise by encoding the quantum data of one qubit into many qubits
  - **Physical qubits**: physical devices like today's qubits
  - **Logical qubits**: an abstraction representing the collection of qubits in a code that act like one high-fidelity qubit

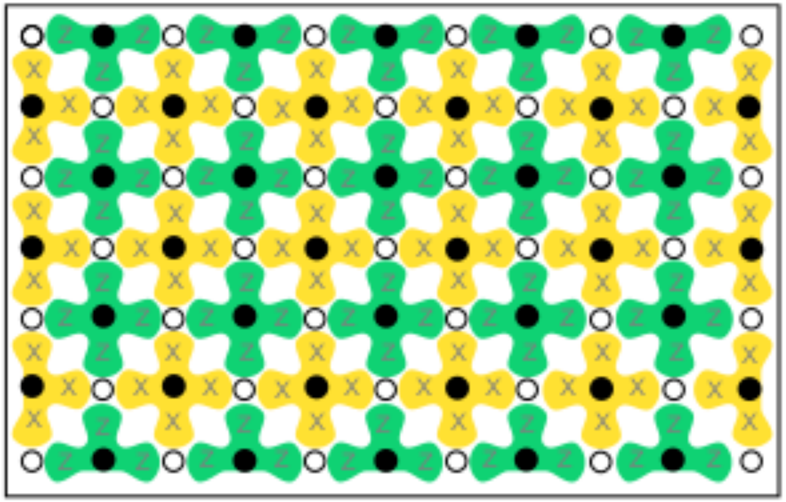
Basic assumption:
1 qubit with error rates a **billion** times better than today

Is much harder than

1000 qubits with error rates **ten** times better than today

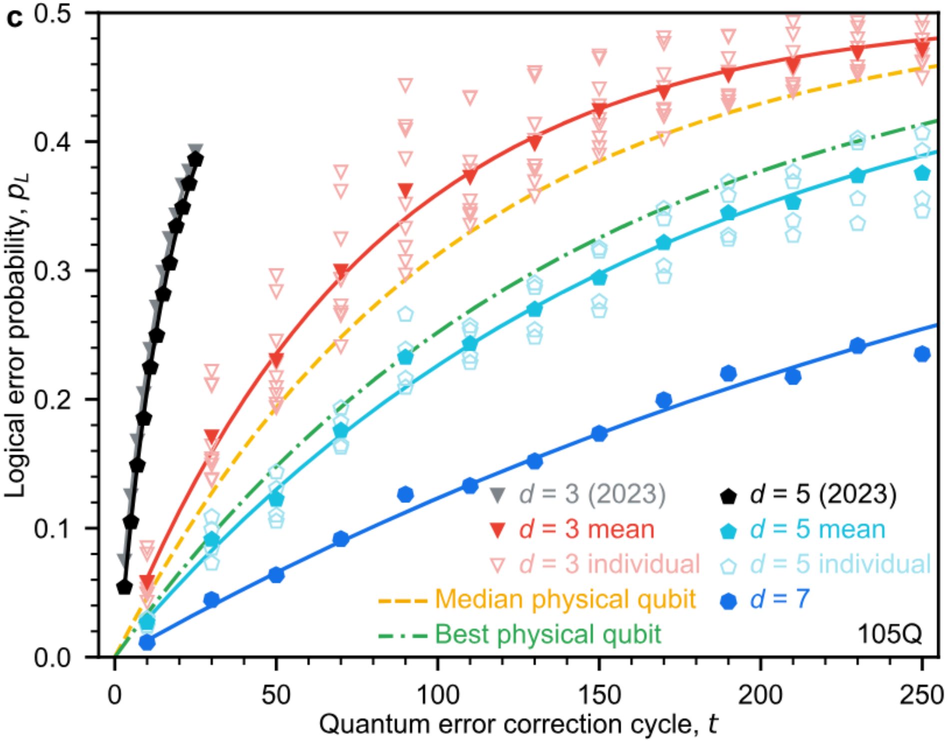
Surface Codes

- Most practical code at the moment
- Uses a 2-dimensional grid of qubits, each connected to its neighbours (easy to build)
- Suppresses errors exponentially in grid width
- Requires repeating cycles of measurement thousands or millions of times per second



Fowler et al., 2012. Towards practical large-scale quantum computation

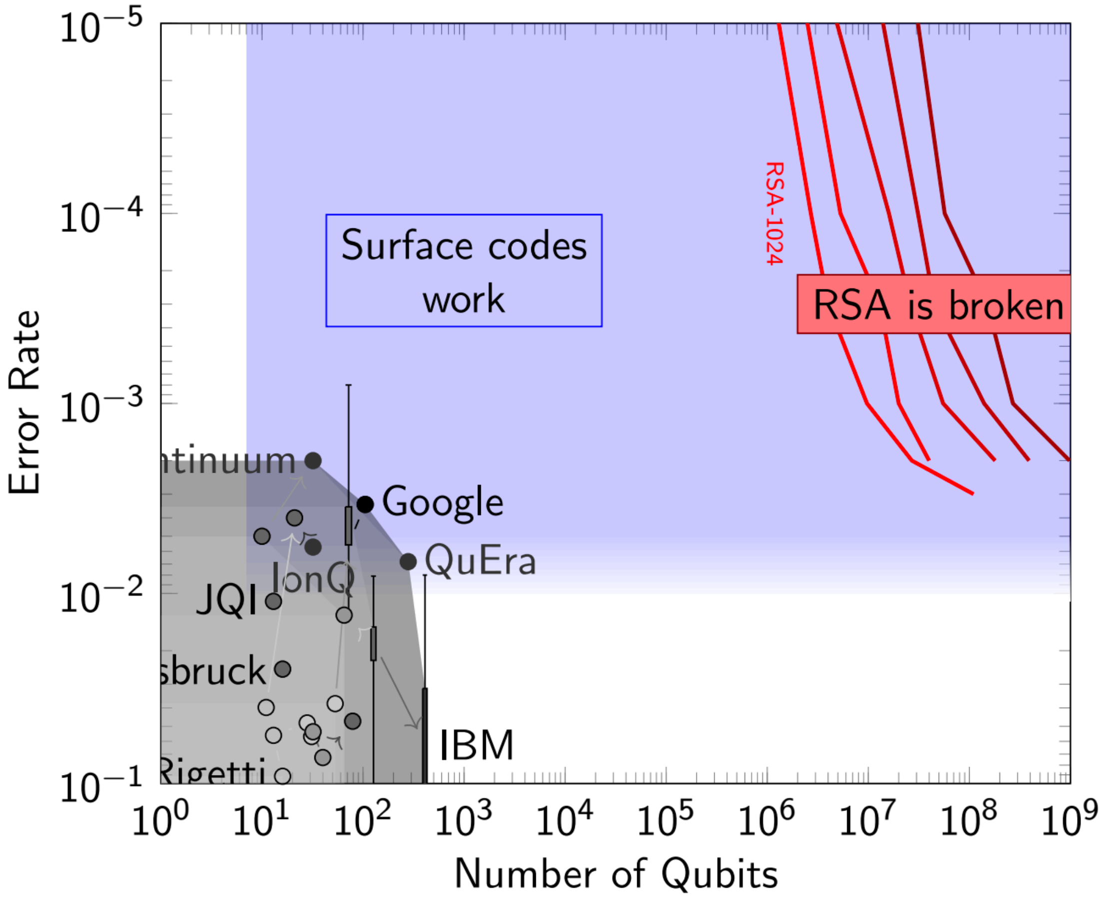
Surface codes today (last week!)



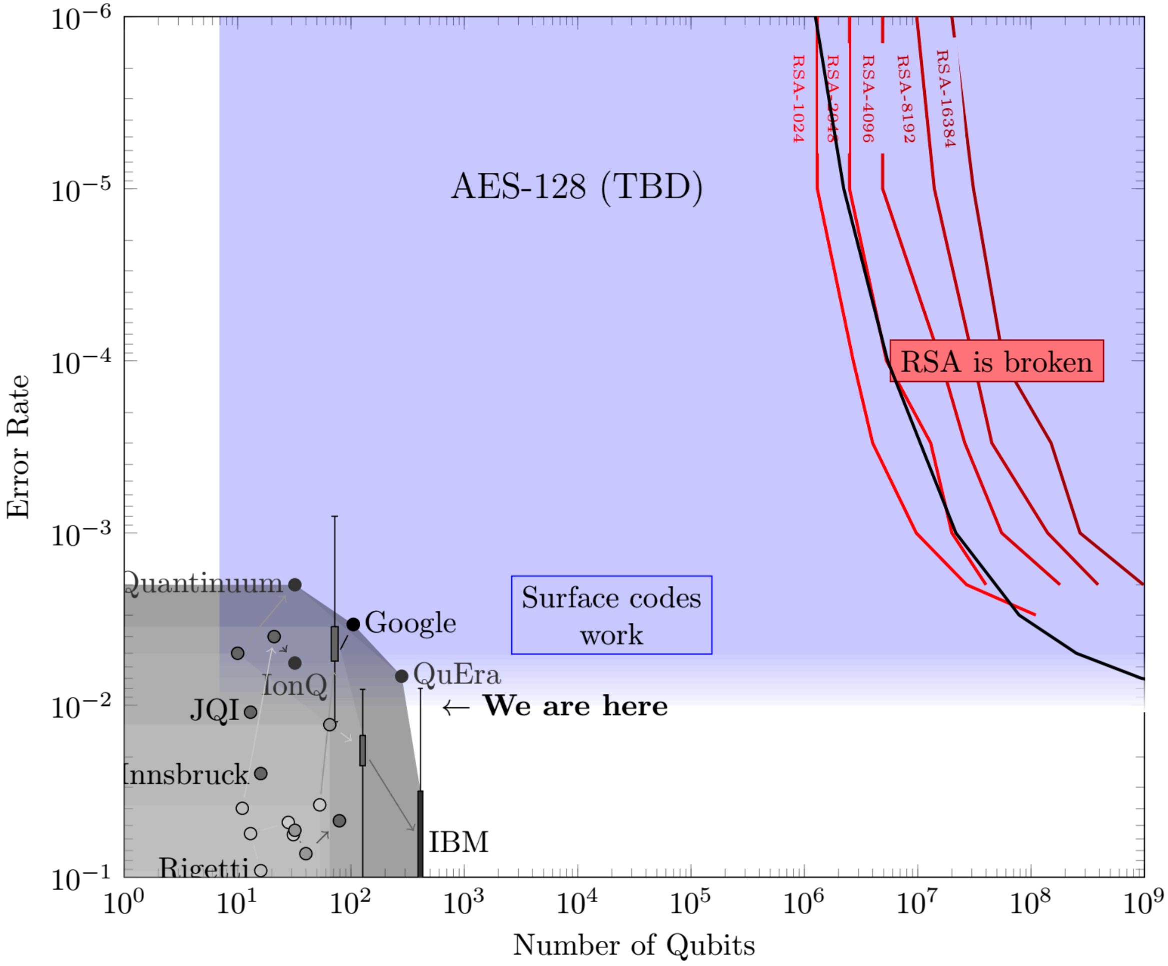
Breakthrough 2024 Experiment from Google Quantum AI:

- Error rate decreases as distance increases
- Logical qubit with smaller errors than physical qubits
- Real-time decoding at 1.1 μs cycle length

Aside: how long to break RSA?



AES is easier to break than RSA!? No



Do not forget runtime!

Error Correction Summary

- **Physical qubits** are the qubits we see today
- **Logical qubits** are the qubits in the circuits we design
- Each logical qubit requires **thousands** of physical qubits
- Correcting errors requires frequent (ex: thousands of times per second) operations on the quantum computer
- The gates we can do on the physical qubits are different than the gates on logical qubits

MISCONCEPTION
#1

~~Misconception: Qubits are the limiting factor for quantum circuits~~

Correct: Even if physical qubits are limiting, “logical qubits” translate into “physical qubits” in a non-trivial way

MISCONCEPTION
#2

Misconception: Because of the square-root speed-up, we should double key sizes

What would a Grover attack look like?

What would a Grover attack look like?

Consider DES. A 56-bit key needs 2^{56} (classical) iterations to break. If each iteration takes 100 clock cycles, than a modern 5 GHz CPU would break DES in…

What would a Grover attack look like?

Consider DES. A 56-bit key needs 2^{56} (classical) iterations to break. If each iteration takes 100 clock cycles, than a modern 5 GHz CPU would break DES in…

… 46 years!?

What would a Grover attack look like?

Consider DES. A 56-bit key needs 2^{56} (classical) iterations to break. If each iteration takes 100 clock cycles, than a modern 5 GHz CPU would break DES in…

… 46 years!?

All realistic attacks are parallel.

MAXDEPTH

MAXDEPTH

In NIST’s 2017 call for post-quantum cryptography, they introduced “MAXDEPTH”, a metric to account for this issue in security analysis. They restricted attacks to one of 3 options:

MAXDEPTH

In NIST's 2017 call for post-quantum cryptography, they introduced "MAXDEPTH", a metric to account for this issue in security analysis. They restricted attacks to one of 3 options:

- 2^{40} logical operations, "the approximate number of gates that presently envisioned quantum computing architectures are expected to serially perform in a year"

MAXDEPTH

In NIST's 2017 call for post-quantum cryptography, they introduced "MAXDEPTH", a metric to account for this issue in security analysis. They restricted attacks to one of 3 options:

- 2^{40} logical operations, "the approximate number of gates that presently envisioned quantum computing architectures are expected to serially perform in a year"
- 2^{64}, "the approximate number of gates that current classical computing architectures can perform serially in a decade"

MAXDEPTH

In NIST's 2017 call for post-quantum cryptography, they introduced "MAXDEPTH", a metric to account for this issue in security analysis. They restricted attacks to one of 3 options:

- 2^{40} logical operations, "the approximate number of gates that presently envisioned quantum computing architectures are expected to serially perform in a year"
- 2^{64}, "the approximate number of gates that current classical computing architectures can perform serially in a decade"
- 2^{96}, "the approximate number of gates that atomic scale qubits with speed of light propagation times could perform in a millennium"

Parallel Attacks

Classical brute-force search does not care about parallelism. Total number of operations stays constant.

> If you're buying server time, you pay for each CPU-hour. Total price to break DES stays the same.

Grover search **does** care about parallelism.

Parallel Grover

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

- Take the key space of 2^n keys, partition into P subsets, each machine searches a different subset

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

- Take the key space of 2^n keys, partition into P subsets, each machine searches a different subset

Now the search space (each subset) has size $\frac{2^n}{P}$. Grover will find the key in the time $O\left(\sqrt{\frac{2^n}{P}}\right)$

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

- Take the key space of 2^n keys, partition into P subsets, each machine searches a different subset

Now the search space (each subset) has size $\frac{2^n}{P}$. Grover will find the key in the time $O\left(\sqrt{\frac{2^n}{P}}\right)$

But the original search was time $O(\sqrt{2^n})$. The time was reduced **only** by a factor of $\sqrt{P}$, not P

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

- Take the key space of 2^n keys, partition into P subsets, each machine searches a different subset

Now the search space (each subset) has size $\frac{2^n}{P}$. Grover will find the key in the time $O\left(\sqrt{\frac{2^n}{P}}\right)$

But the original search was time $O(\sqrt{2^n})$. The time was reduced **only** by a factor of $\sqrt{P}$, not P

Worse: total cost (# operations) has gone **up** to $P \times O(\sqrt{2^n/P}) = O(\sqrt{P2^n})$

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Parallel Grover

Best method to parallelize Grover to P machines:

- Ta
  m

Now t e

key in

But

only

Worse: total cost (# operations) has gone **up** to

$$P \times O(\sqrt{2^n/P}) = O(\sqrt{P2^n})$$

Main takeaway:

Grover parallelizes badly.

Zalka. Grover's quantum searching algorithm is optimal. 1997.

Common misconception: Decoherence

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

NIST's limit **does not** reflect decoherence concerns.

Common misconception: Decoherence

Today’s qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

NIST’s limit **does not** reflect decoherence concerns.

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

NIST's limit **does not** reflect decoherence concerns.

Quantum error correction lets me take any qubit which stays coherent for time T, and create an encoded qubit out of C such qubits which stays coherent for time $T \times \exp(\sqrt{C})$

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

NIST's limit **does not** reflect decoherence concerns.

Quantum error correction lets me take any qubit which stays coherent for time T, and create an encoded qubit out of C such qubits which stays coherent for time $T \times \exp(\sqrt{C})$

Common misconception: Decoherence

Today's qubits last only a fraction of a second before **decohering**, i.e., losing their quantum data

NIST's limit **does not** reflect decoherence concerns.

Quantum error correction lets me take any qubit which stays coherent for time T, and create an encoded qubit out of C such qubits which stays coherent for time $T \times \exp(\sqrt{C})$

The real constraint: Secrets are not valuable forever

MISCONCEPTION
#2

~~Misconception: Because of the square-root speed-up, we should double key sizes~~

~~Correct~~: My opinion: Parallel Grover attacks are so expensive we will not see them break AES-128 our lifetimes, and possibly never at all.

MISCONCEPTION
#3

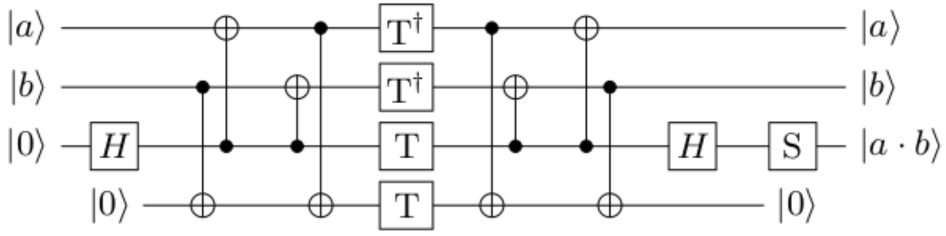
Misconception: Breaking AES-128 will take $2^{64} \times$ (small constant) quantum time, where the small constant is well-known

QUANTUM CIRCUIT DESIGN

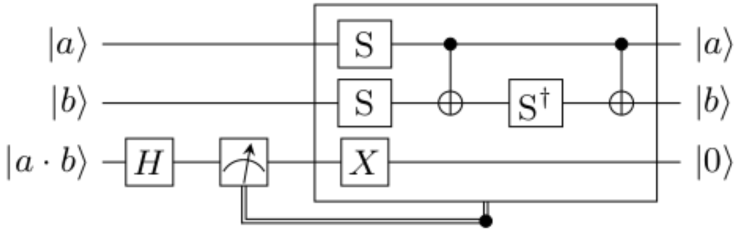
A crash course

What is a quantum circuit?

- A quantum circuit is a list of which gates to apply, to which qubits, in what order



(a) AND gate.



(b) AND† gate.

Gates on error corrected codes

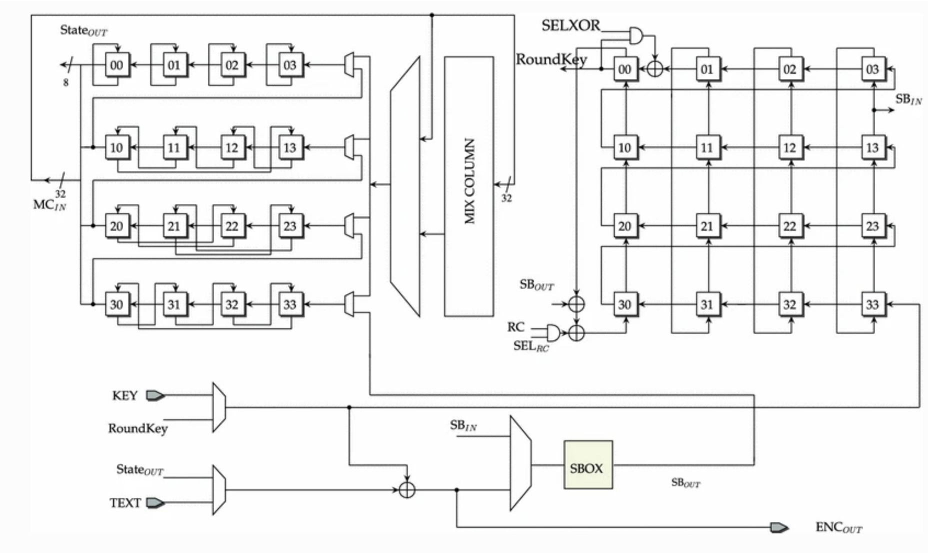
Many different equivalent gate sets are possible

Typically we consider a gate set called “Clifford + T”. Why?

- Any quantum operation can be approximated with Clifford + T gates
- Clifford gates are easy to apply on a surface code
- T gates are **not** easy and require “magic states”

For this reason we often emphasize T gates when designing circuits

Grover Iterations

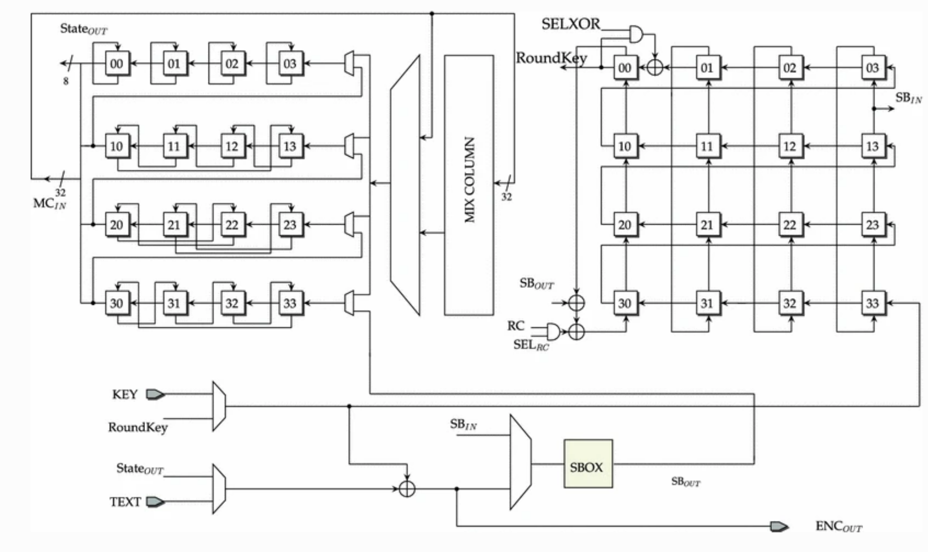


Banik, Bogdanov, Regazzoni. Compact circuits for combined AES encryption/decryption. JCE 2017

*certain quantum tricks can avoid this

Grover Iterations

- Quantum theory states that any classical circuit can be transformed to a quantum circuit with polynomial overhead. Simple as this?

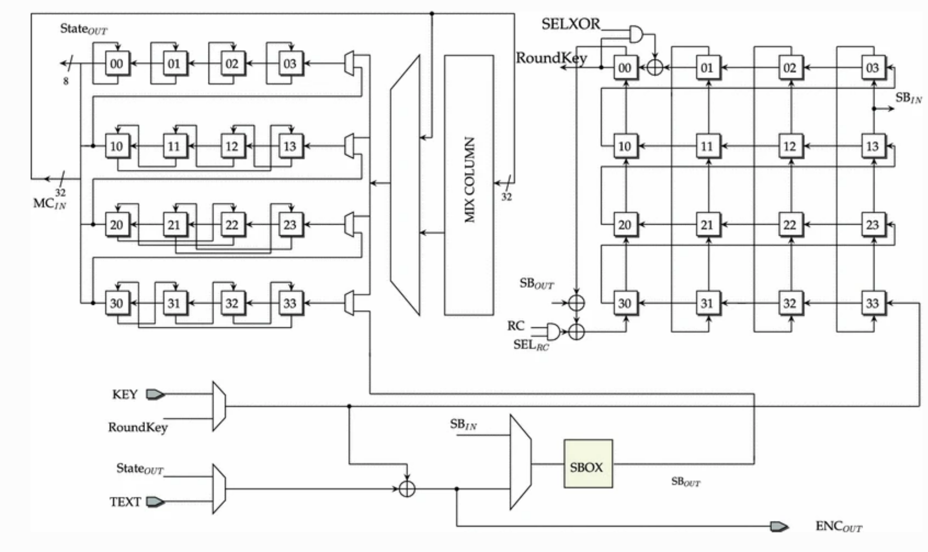


Banik, Bogdanov, Regazzoni. Compact circuits for combined AES encryption/decryption. JCE 2017

*certain quantum tricks can avoid this

Grover Iterations

- Quantum theory states that any classical circuit can be transformed to a quantum circuit with polynomial overhead. Simple as this?
- Quantum circuits must be **constant time** and **reversible*** . This adds noticeable overhead!

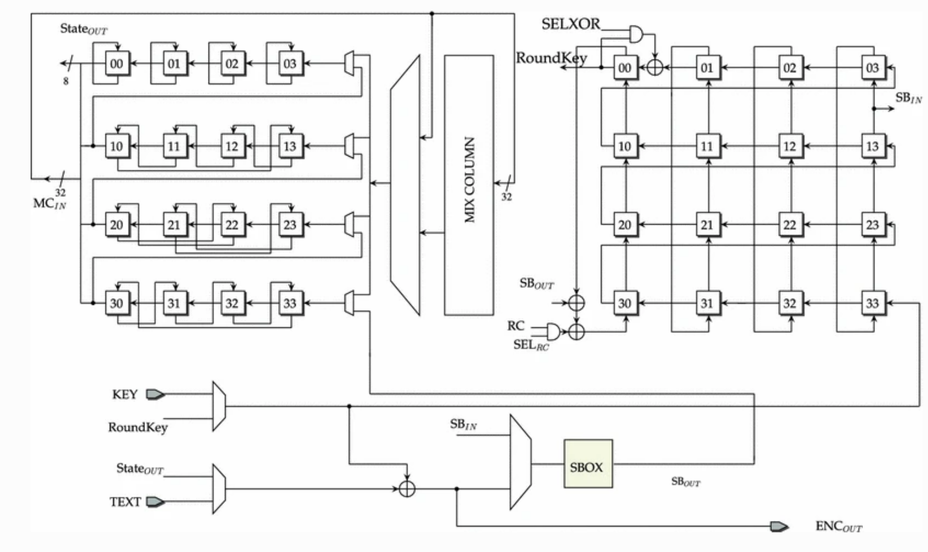


Banik, Bogdanov, Regazzoni. Compact circuits for combined AES encryption/decryption. JCE 2017

*certain quantum tricks can avoid this

Grover Iterations

- Quantum theory states that any classical circuit can be transformed to a quantum circuit with polynomial overhead. Simple as this?
- Quantum circuits must be **constant time** and **reversible*** . This adds noticeable overhead!
- How do we optimize our quantum circuits? Number of qubits, runtime/depth, number of gates…?

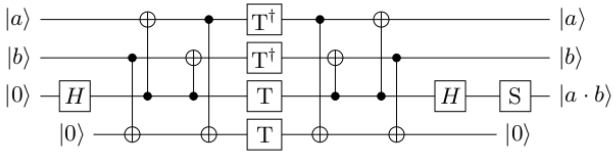


Banik, Bogdanov, Regazzoni. Compact circuits for combined AES encryption/decryption. JCE 2017

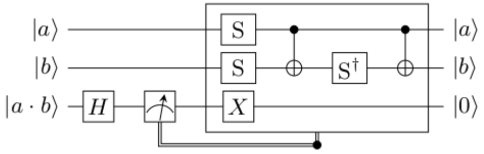
*certain quantum tricks can avoid this

Quantum Circuit Design

Low-level quantum circuits look like this:

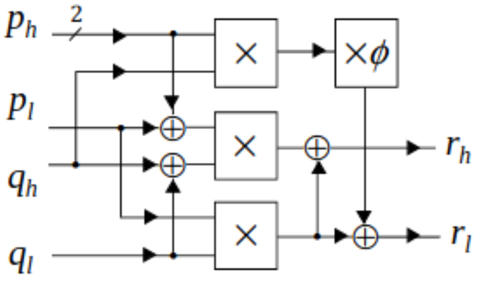


(a) AND gate.



(b) AND† gate.

AES circuits look like this:

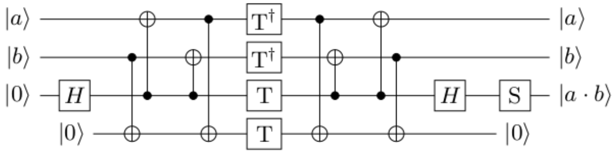


Standard practice: design reversible classical circuit (XOR, AND, etc.), translate to quantum gates (X, CNOT, Toffoli), translate these to Clifford+T

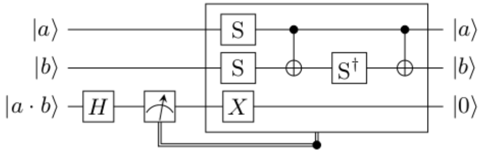
Diagrams from Chung, Lee, Choi, Lee. Alternative Tower Field Construction for Quantum Implementation of the AES S-box. TC 2020

Quantum Circuit Design

Low-level quantum circuits look like this:

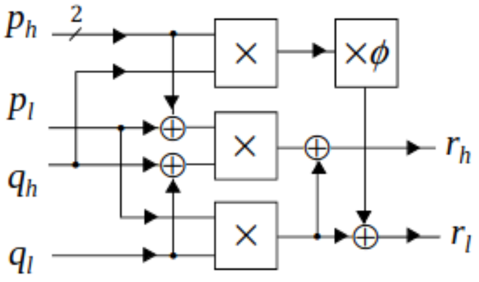


(a) AND gate.



(b) AND† gate.

AES circuits look like this:



Standard practice: design reversible classical circuit (XOR, AND, etc.), translate to quantum gates (X, CNOT, Toffoli), translate these to Clifford+T

Important but confusing: Toffoli gates are not T gates! But Toffoli is the only gate whose Clifford+T circuit needs T gates

Diagrams from Chung, Lee, Choi, Lee. Alternative Tower Field Construction for Quantum Implementation of the AES S-box. TC 2020

Optimize for total gates?

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)
- For future codes, who knows?

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)
- For future codes, who knows?

Two important facts:

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)
- For future codes, who knows?

Two important facts:

- No matter the error-correcting code, at least one gate is difficult (Eastin-Knill theorem)

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)
- For future codes, who knows?

Two important facts:

- No matter the error-correcting code, at least one gate is difficult (Eastin-Knill theorem)
- Any gate set can be converted to another with $O(1)$ overhead

Optimize for total gates?

Each gate might be costly (computation, energy for a laser, etc.)

But which gates to count?

- For surface-code error-correction logical qubits, we have Clifford + T (with T gates **much** harder)
- For future codes, who knows?

Two important facts:

- No matter the error-correcting code, at least one gate is difficult (Eastin-Knill theorem)
- Any gate set can be converted to another with $O(1)$ overhead

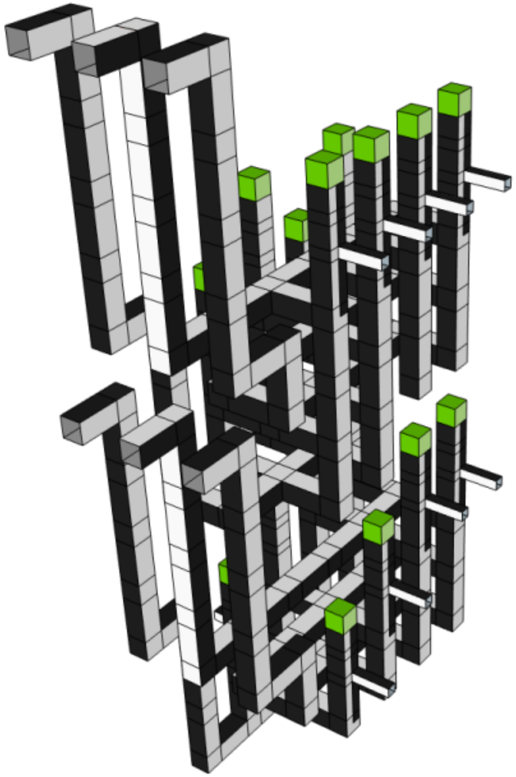
So why engage in this exercise at all?

Optimize for Toffoli count?

Certain gates look classical:

- X is like a NOT gate
- CNOT is like an XOR gate
- Toffoli is like an AND gate

Toffoli can simulate the others, so more conservative to expect Toffoli is hard



Two Toffolis in a surface code.

From: Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019.

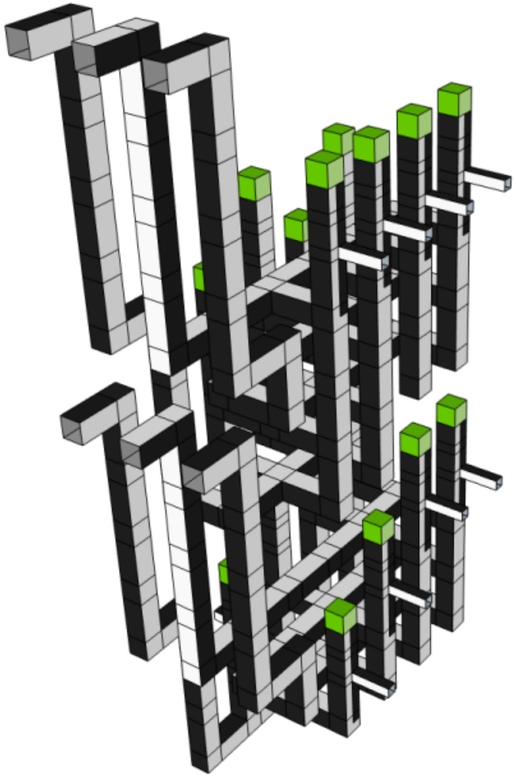
Optimize for Toffoli count?

Certain gates look classical:

- X is like a NOT gate
- CNOT is like an XOR gate
- Toffoli is like an AND gate

Toffoli can simulate the others, so more conservative to expect Toffoli is hard

But! Modern quantum techniques break away from reversible classical computing!



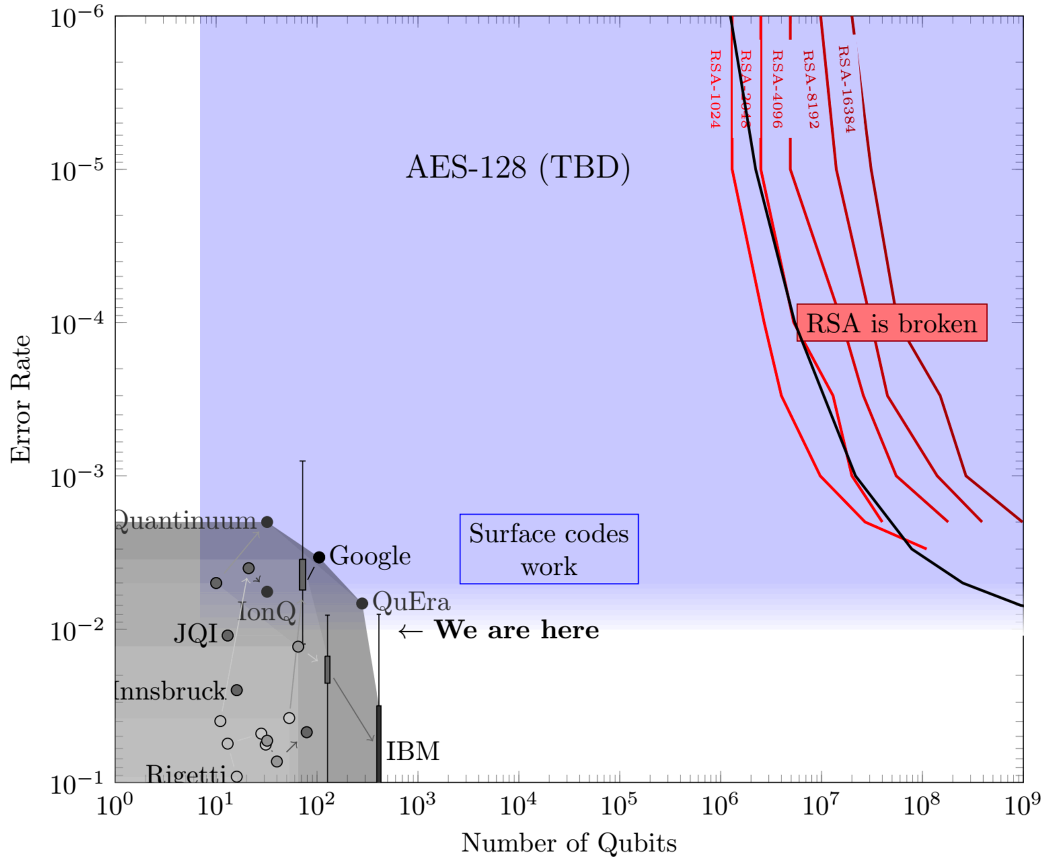
Two Toffolis in a surface code.

From: Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019.

Optimize for depth? Or depth x width?

- Since a single thread of Grover doesn't need many qubits, we must optimize total execution speed

- Or: focus on depth x width. Like area-time, but could reflect error correction overhead, or opportunity costs



Other metrics

- Since Grover's algorithm parallelizes badly, a shorter-depth AES subroutine has a disproportionate impact on total operation count. Thus:

  - If we want to optimize gate cost of the **overall** attack, we should optimize **gates x depth** for the AES circuit itself
  - If we want to optimize depth x width cost of the **overall** attack, we should optimize $\mathbf{depth}^2 \times \mathbf{width}$ for the AES circuit itself

We noticed this and optimized for it in 2020*; the best such circuits today are from Jang et al. "Quantum Analysis of AES".

*Jaques, Naehrig, Roetteler, Virdia. Implementing Grover oracles for quantum key search on AES and LowMC. Eurocrypt 2020.

Doubts about AES circuits

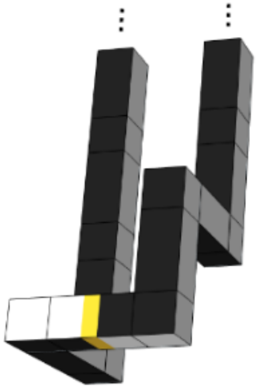
The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:

Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:

Qubits require space to move around

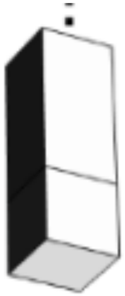


Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:

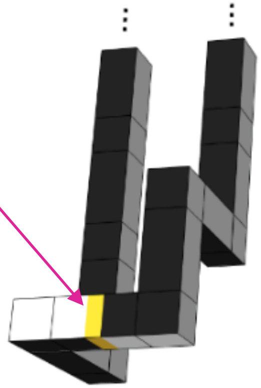
X gates are compiled away entirely



Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:



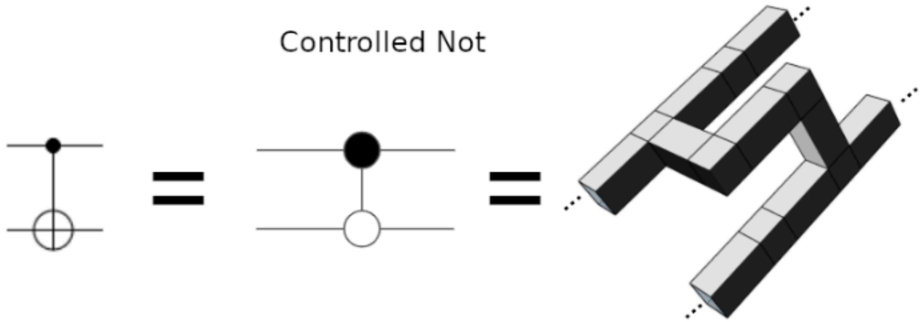
H gates are nearly free

Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:

CNOT gates require complicated "piping"

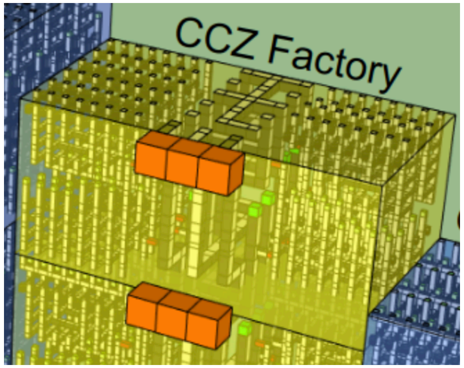


Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

The circuits previously described use the Clifford+T gate set. Clifford + T is a natural choice for surface codes. But in an actual surface code:

Toffoli gates require ENORMOUS “factories”



Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

Doubts about AES circuits

Doubts about AES circuits

- The logical circuits ignore difficulties and subtleties of the surface code

Doubts about AES circuits

- The logical circuits ignore difficulties and subtleties of the surface code
- However, the circuits are based on a gate set justified by the surface code

Doubts about AES circuits

- The logical circuits ignore difficulties and subtleties of the surface code
- However, the circuits are based on a gate set justified by the surface code

Doubts about AES circuits

- The logical circuits ignore difficulties and subtleties of the surface code
- However, the circuits are based on a gate set justified by the surface code

- If surface codes continue to dominate: the cost estimates are incomplete

Doubts about AES circuits

- The logical circuits ignore difficulties and subtleties of the surface code
- However, the circuits are based on a gate set justified by the surface code

- If surface codes continue to dominate: the cost estimates are incomplete
- If surface codes are replaced: the circuits were likely optimized for the wrong gate set

Why didn’t we make surface code layouts for AES?

Why didn't we make surface code layouts for AES?

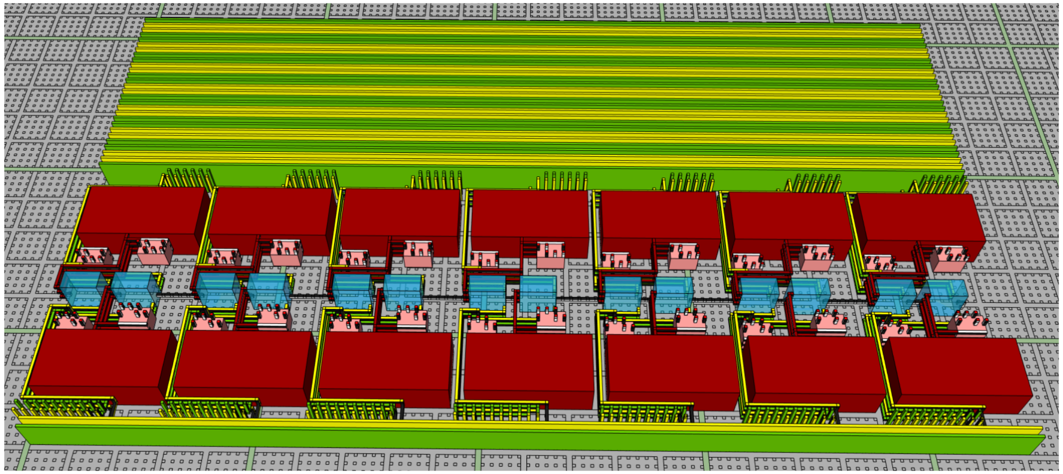
- The good reason: it would be premature to assume surface codes will be the dominant quantum architecture

Why didn’t we make surface code layouts for AES?

- The good reason: it would be premature to assume surface codes will be the dominant quantum architecture
- The real reason: no good tools existed to work with surface code layouts

Why didn’t we make surface code layouts for AES?

- The good reason: it would be premature to assume surface codes will be the dominant quantum architecture
- The real reason: no good tools existed to work with surface code layouts
- All of the diagrams I’ve shown were made “by hand” in SketchUp 💀

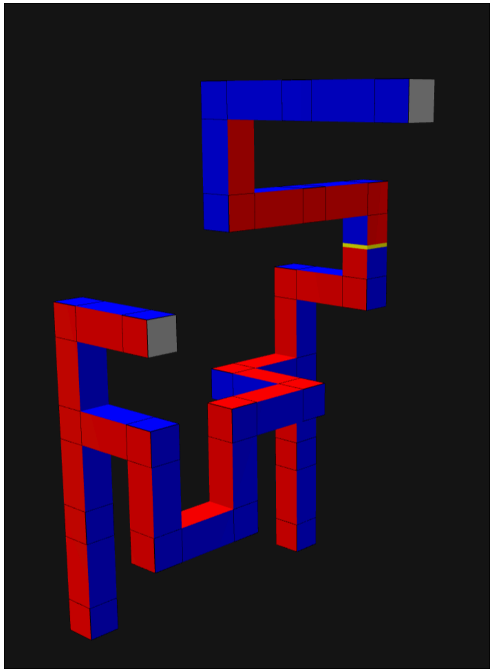


Diagrams from Gidney and Fowler. Flexible layout of surface code computations using AutoCCZ states. 2019

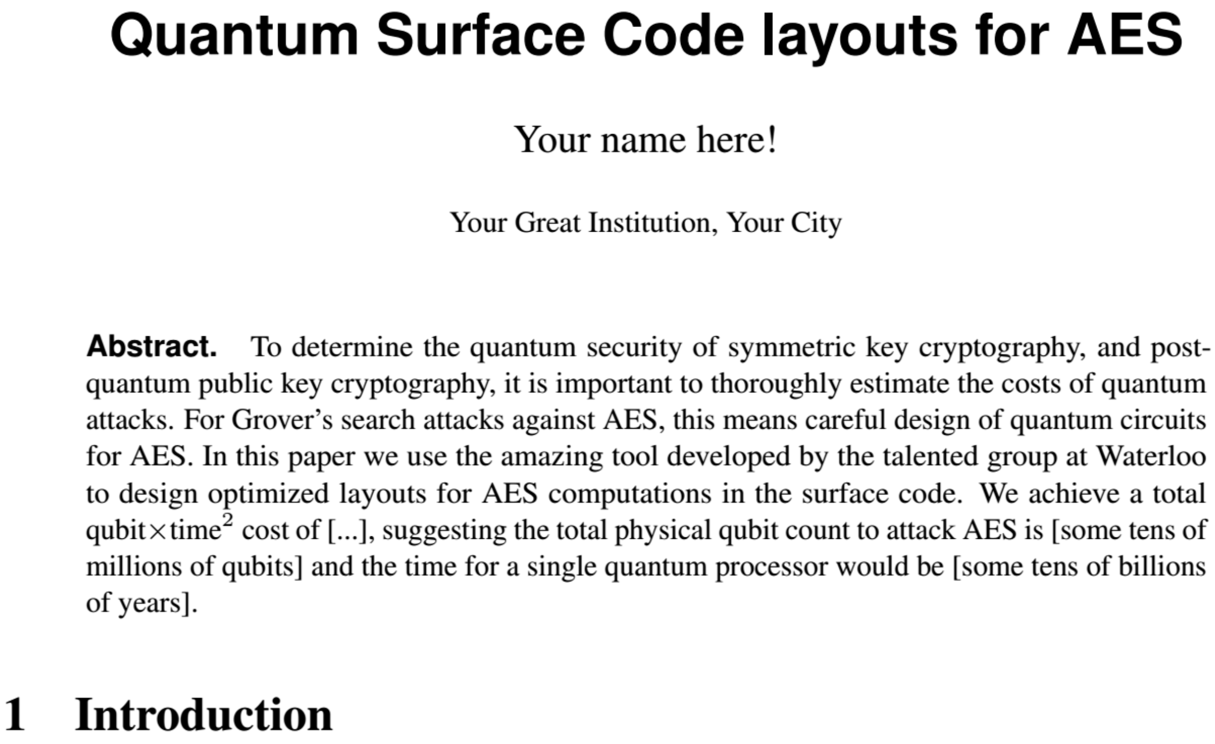
Good news: a new tool exists!

- Myself and Grace Terhljan adapted a tool from Tan, Niu, Gidney:

```
from  lassir_generator import lassir_gen
from lattice_surgery_compiler import LatticeSurgerySolution
from qubit_move_plot import Qubit, cnot, path_find_move


test_lassir = lassir_gen(10,10,10)
test = LatticeSurgerySolution(lassir=dict())
test.load_lassir("olssco/10x10x10_blank.lassir")

first_qubit = Qubit([3,3,3],[], test.lassir, False, orientation = 0)
second_qubit = Qubit([6,3,3],[], test.lassir, False, orientation = 0)

first_qubit.move_z([3,3,6])
second_qubit.move_z([6,3,6])

first_qubit.move_y([3,4,6])
second_qubit.move_y([6,4,6])

cnot(first_qubit, [3,3,6], second_qubit, [6,3,6])

first_qubit.hadamard()
first_qubit.move_z([3,3,7])

path_find_move(first_qubit, [5,6,8],[0,1])


test.to_3d_model_gltf("ex_for_talk.gltf")
```



To appear in CHES 2026:



Quantum Surface Code layouts for AES

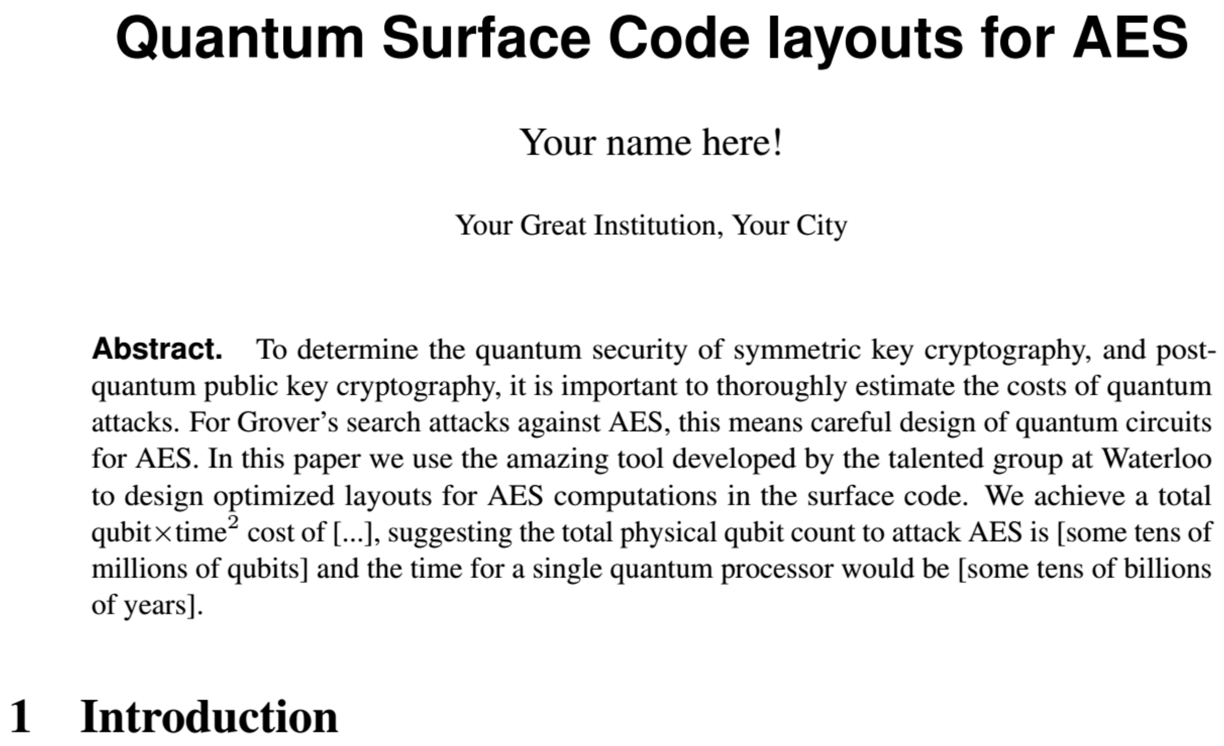
Your name here!

Your Great Institution, Your City

Abstract. To determine the quantum security of symmetric key cryptography, and post-quantum public key cryptography, it is important to thoroughly estimate the costs of quantum attacks. For Grover's search attacks against AES, this means careful design of quantum circuits for AES. In this paper we use the amazing tool developed by the talented group at Waterloo to design optimized layouts for AES computations in the surface code. We achieve a total qubit×time^2 cost of [...], suggesting the total physical qubit count to attack AES is [some tens of millions of qubits] and the time for a single quantum processor would be [some tens of billions of years].

1 Introduction

To appear in CHES 2026:



Quantum Surface Code layouts for AES

Your name here!

Your Great Institution, Your City

Abstract. To determine the quantum security of symmetric key cryptography, and post-quantum public key cryptography, it is important to thoroughly estimate the costs of quantum attacks. For Grover's search attacks against AES, this means careful design of quantum circuits for AES. In this paper we use the amazing tool developed by the talented group at Waterloo to design optimized layouts for AES computations in the surface code. We achieve a total qubit$\times$time^2 cost of [...], suggesting the total physical qubit count to attack AES is [some tens of millions of qubits] and the time for a single quantum processor would be [some tens of billions of years].

1 Introduction

Let's not be so preoccupied with whether we **could** write this paper that we forget to ask whether **should** write this paper

Why not make surface code layouts?

From Jang et al. Quantum Analysis of AES: Lowering the limit of Quantum Attack Complexity. 2022.

Key Size	**Allowed Depth**	**Total Gate Cost**	**Total Logical Qubit Count**
128 bits	2^{40}	2^{116}	2^{80}
192 Bits	2^{40}	2^{182}	2^{145}
256 Bits	2^{40}	2^{246}	2^{209}

Why not make surface code layouts?

From Jang et al. Quantum Analysis of AES: Lowering the limit of Quantum Attack Complexity. 2022.

Key Size	Allowed Depth	Total Gate Cost	Total Logical Qubit Count
128 bits	2^{40}	2^{116}	2^{80}
192 Bits	2^{40}	2^{182}	2^{145}
256 Bits	2^{40}	2^{246}	2^{209}

For a surface code big enough for this computation, each logical depth x qubit operation requires $2 \times 36^3 = 2^{16}$ operations.

Why not make surface code layouts?

From Jang et al. Quantum Analysis of AES: Lowering the limit of Quantum Attack Complexity. 2022.

Key Size	Allowed Depth	Total Gate Cost	Total Logical Qubit Count
128 bits	2^{40}	2^{116}	2^{80}
192 Bits	2^{40}	2^{182}	2^{145}
256 Bits	2^{40}	2^{246}	2^{209}

For a surface code big enough for this computation, each logical depth x qubit operation requires $2 \times 36^3 = 2^{16}$ operations.

Why not make surface code layouts?

From Jang et al. Quantum Analysis of AES: Lowering the limit of Quantum Attack Complexity. 2022.

Key Size	Allowed Depth	Total Gate Cost	Total Logical Qubit Count
128 bits	2^{40}	2^{116}	2^{80}
192 Bits	2^{40}	2^{182}	2^{145}
256 Bits	2^{40}	2^{246}	2^{209}

For a surface code big enough for this computation, each logical depth x qubit operation requires $2 \times 36^3 = 2^{16}$ operations.

Total quantum operations on a surface code: **at least** 2^{136}

Why not make surface code layouts?

From Jang et al. Quantum Analysis of AES: Lowering the limit of Quantum Attack Complexity. 2022.

Key Size	Allowed Depth	Total Gate Cost	Total Logical Qubit Count
128 bits	2^{40}	2^{116}	2^{80}
192 Bits	2^{40}	2^{182}	2^{145}
256 Bits	2^{40}	2^{246}	2^{209}

For a surface code big enough for this computation, each logical depth x qubit operation requires $2 \times 36^3 = 2^{16}$ operations.

Total quantum operations on a surface code: **at least** 2^{136}
Total classical operations to break AES: 2^{143}

Physical Bounds

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- Landauer's law: any non-reversible operation requires $k_B T \ln(2)$ Joules of energy

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- Landauer's law: any non-reversible operation requires $k_B T \ln(2)$ Joules of energy
- Error-correction operations are non-reversible

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- Landauer's law: any non-reversible operation requires $k_B T \ln(2)$ Joules of energy
- Error-correction operations are non-reversible
- Thus, breaking AES-128 in MAXDEPTH= 2^{40} requires — at the minimum physically possible — 2^{55} Joules of energy

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- Landauer's law: any non-reversible operation requires $k_B T \ln(2)$ Joules of energy
- Error-correction operations are non-reversible
- Thus, breaking AES-128 in MAXDEPTH= 2^{40} requires — at the minimum physically possible — 2^{55} Joules of energy
  - This is the output of an entire nuclear power plant for 1 year

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- Landauer's law: any non-reversible operation requires $k_B T \ln(2)$ Joules of energy
- Error-correction operations are non-reversible
- Thus, breaking AES-128 in MAXDEPTH= 2^{40} requires — at the minimum physically possible — 2^{55} Joules of energy
  - This is the output of an entire nuclear power plant for 1 year
  - Almost certainly the real energy will be orders of magnitude larger

Image: Wikipedia user Heinz-Josef Lücking

Physical Bounds

- If each qubit is 2 microns wide, the 2^{80} qubits necessary would cover the surface of the moon

Image: Wikipedia user Achituv

Physical Bounds

AES-128 can be broken at (logical) cost “only” 2^{89} with MAXDEPTH= 2^{96}. But recall NIST’s reasoning:

> 2^{96} = “the approximate number of gates that atomic scale qubits with speed of light propagation times could perform in a millennium”

Physical Bounds

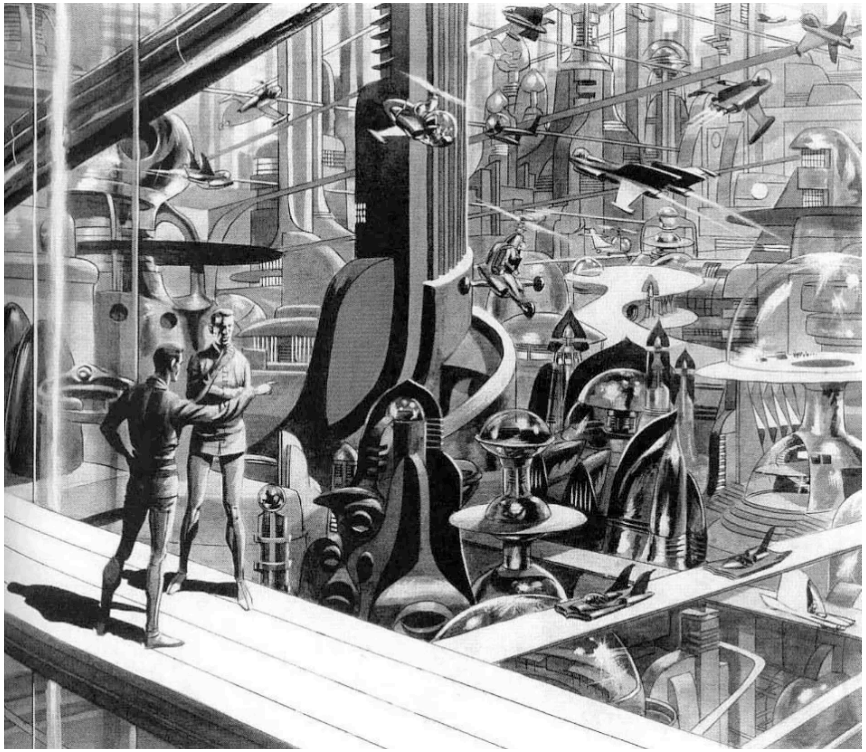
AES-128 can be broken at (logical) cost “only” 2^{89} with MAXDEPTH= 2^{96}. But recall NIST’s reasoning:

> 2^{96} = “the approximate number of gates that atomic scale qubits with speed of light propagation times could perform in a millennium”

This will never be built

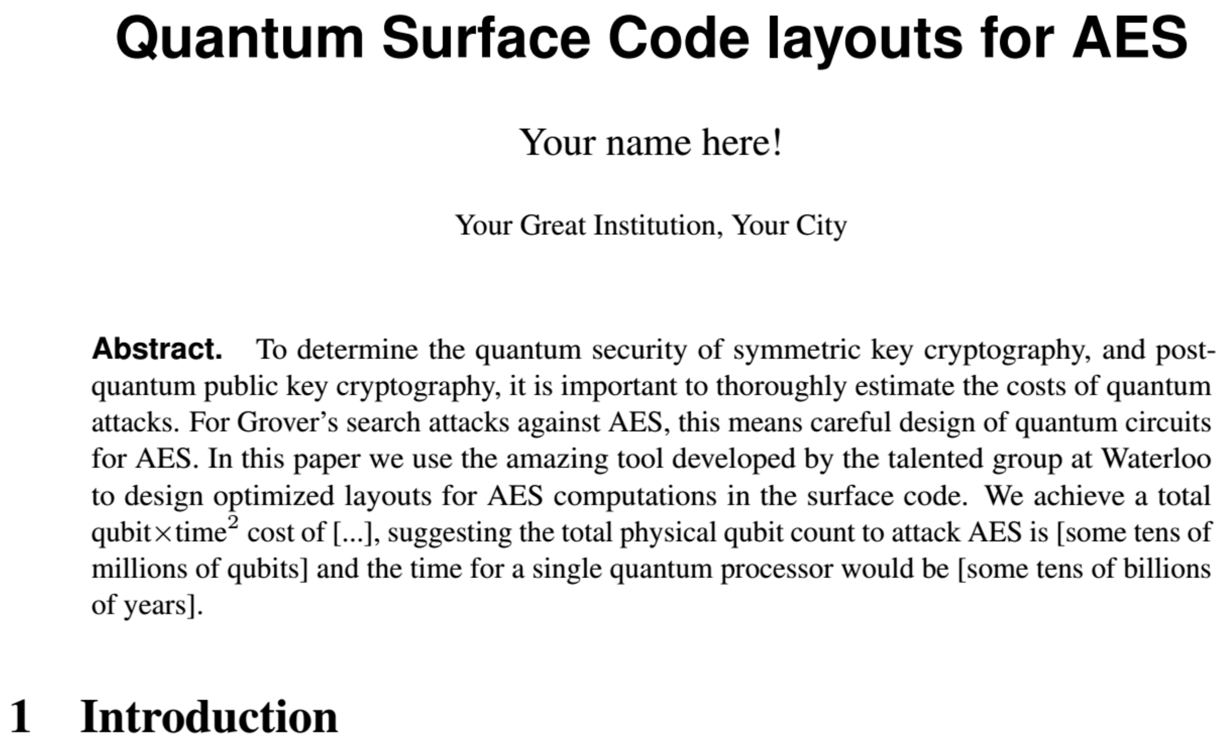
Galactic Algorithms and Science Fiction

We can talk about computers on Dyson spheres and black hole computers and harnessing supernovae, but let's be real.



Illustrator: Wally Wood

To appear in CHES 2026:



Quantum Surface Code layouts for AES

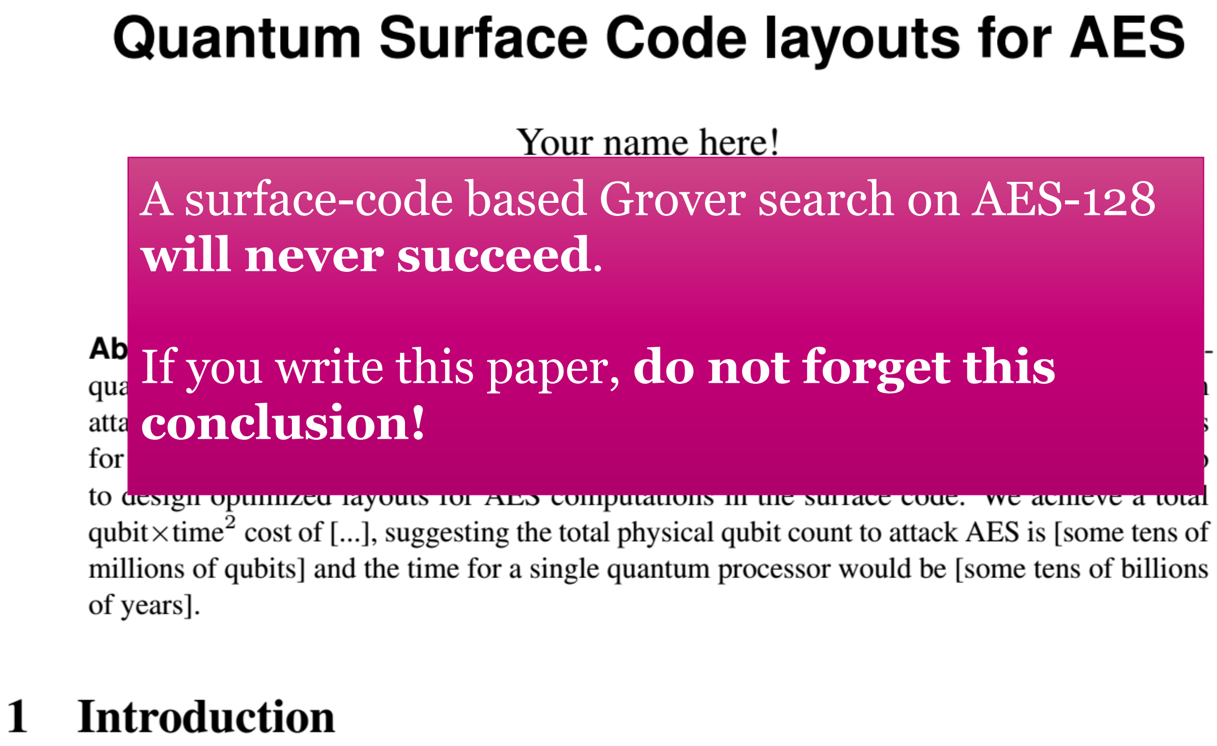
Your name here!

Your Great Institution, Your City

Abstract. To determine the quantum security of symmetric key cryptography, and post-quantum public key cryptography, it is important to thoroughly estimate the costs of quantum attacks. For Grover's search attacks against AES, this means careful design of quantum circuits for AES. In this paper we use the amazing tool developed by the talented group at Waterloo to design optimized layouts for AES computations in the surface code. We achieve a total qubit$\times$time^2 cost of [...], suggesting the total physical qubit count to attack AES is [some tens of millions of qubits] and the time for a single quantum processor would be [some tens of billions of years].

1 Introduction

To appear in CHES 2026:



Quantum Surface Code layouts for AES

Your name here!

Ab

qua

atta

for

to design optimized layouts for AES computations in the surface code. We achieve a total qubit×time^2 cost of [...], suggesting the total physical qubit count to attack AES is [some tens of millions of qubits] and the time for a single quantum processor would be [some tens of billions of years].

1 Introduction

A surface-code based Grover search on AES-128 **will never succeed**.

If you write this paper, **do not forget this conclusion!**

MISCONCEPTION
#3

~~Misconception: Breaking AES-128 will take 2^{64} × (small constant) quantum time, where the small constant is well-known~~

Correct: Parallelism means it is not 2^{64}; future architectures are too uncertain to have good circuit designs

CONCLUSIONS

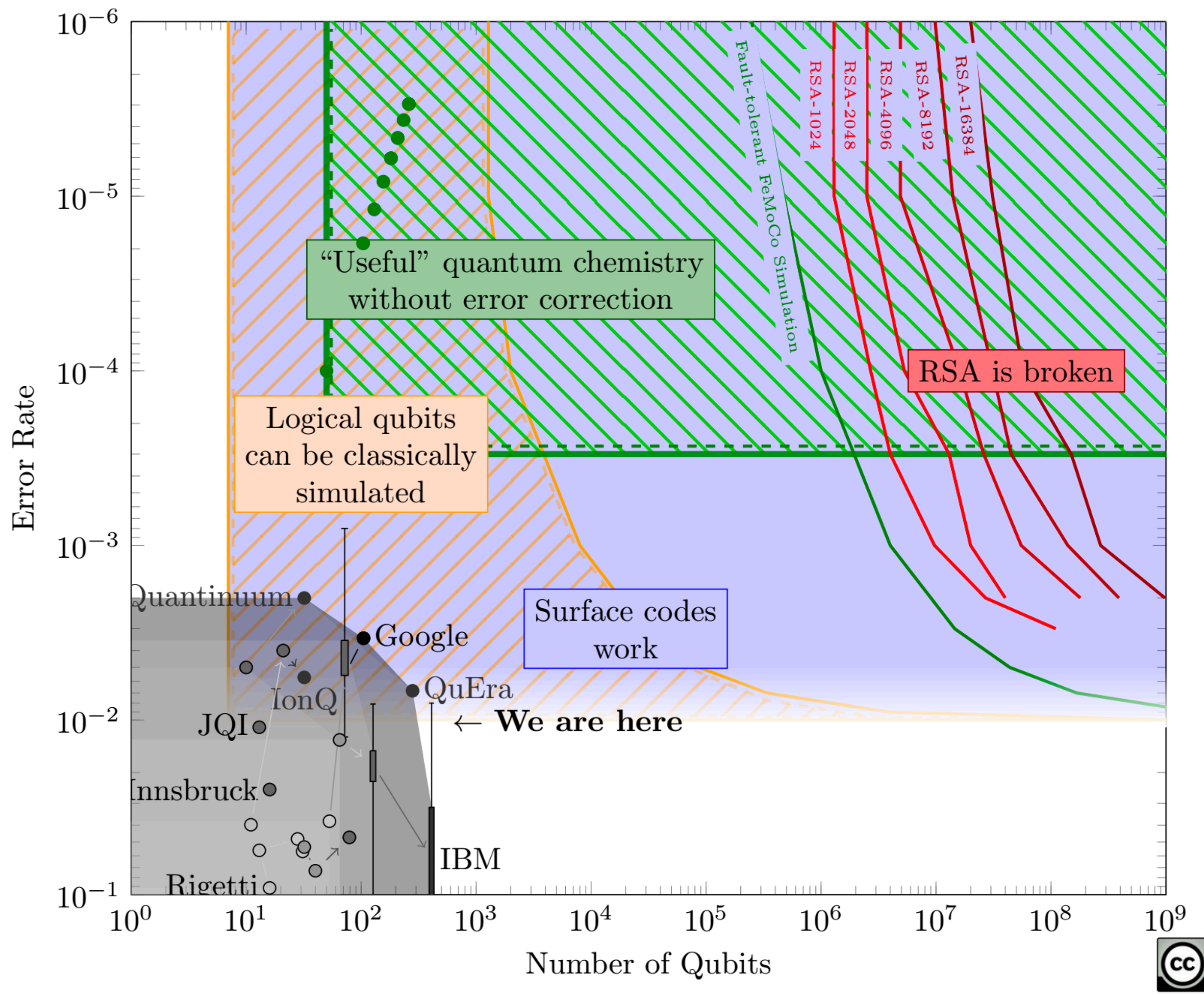
- Physical and logical qubits are different things
- Grover's algorithm parallelizes badly
- It is hard (pointless, even!) to guess now that the optimal AES circuit will be, since technology changes
- AES-128 is probably safe from classical and quantum attacks in our lifetimes

CONCLUSIONS

- Physical and logical qubits are different things
- Grover's algorithm parallelizes badly
- It is hard (pointless, even!) to guess now that the optimal AES circuit will be, since technology changes
- AES-128 is probably safe from classical and quantum attacks in our lifetimes

Thank you, I'm done talking now

Samuel Jaques


Error Rate
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
10^{7}
10^{8}
10^{9}
Number of Qubits
Fault-tolerant FeMoCo Simulation
RSA-1024
RSA-2048
RSA-4096
RSA-8192
RSA-16384
"Useful" quantum chemistry without error correction
RSA is broken
Logical qubits can be classically simulated
Surface codes work
Quantinuum
Google
IonQ
QuEra
JQI
← **We are here**
Innsbruck
IBM
Rigetti