Factoring with a Quantum Computer: The State of the Art

Seyoon Ragavan
MIT

Part 1: A History of Quantum Factoring

Part 2: Factoring Some Integers in Sublinear Quantum Space

ARTIFICIAL INTELLIGENCE • STARTUPS • VENTURE

Quantum Computing Funding Hits Record High With Apparent AI Boost

Chris Metinko November 14, 2024

Sundar Pichai
@sundarpichai

Introducing Willow, our new state-of-the-art quantum computing chip with a breakthrough that can reduce errors exponentially as we scale up using more qubits, cracking a 30-year challenge in the field. In benchmark tests, Willow solved a standard computation in <5 mins that would take a leading supercomputer over 10^25 years, far beyond the age of the universe(!).

10:36 PM · Dec 9, 2024 · **18.8M** Views

Department of Energy Announces $65 Million for Quantum Computing Research

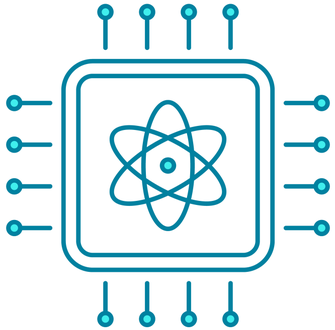
Projects span software, control systems, and algorithms for quantum computing

Office of Science

September 9, 2024

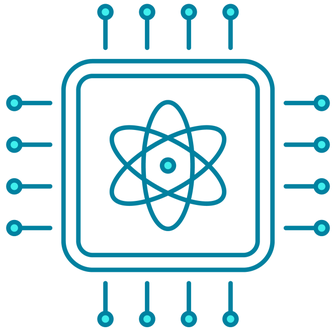
IBM's Quantum Heron Processor breaks speed records with 50x performance boost

Q: Why all this buzz around quantum computing?



Q: Why all this buzz around quantum computing?

*A: Quantum computers **could** solve **some** problems that we believe to be completely intractable with a classical computer.*



Integer Factorisation

Quantum computing's poster child

Given an n-bit integer $N < 2^n$, find its prime factorisation in $\mathrm{poly}(n)$ time.

- Example: $35 \to 5 \times 7$

Integer Factorisation

Quantum computing's poster child

Given an n-bit integer $N < 2^n$, find its prime factorisation in poly(n) time.

- Example: $35 \rightarrow 5 \times 7$
- Fastest known classical algorithms: $2^{\tilde{O}(n^{1/3})}$ time (Pollard 1988; Buhler-Lenstra-Pomerance 1993)

Integer Factorisation

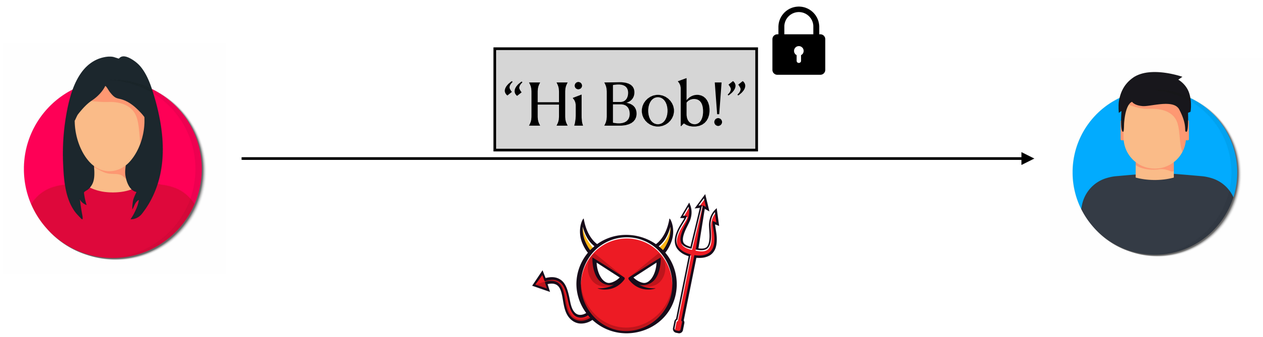
Quantum computing's poster child

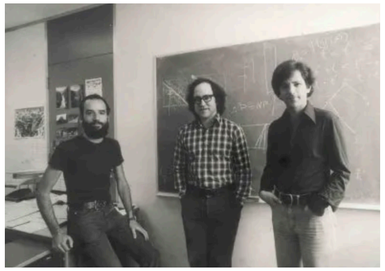
Given an n-bit integer $N < 2^n$, find its prime factorisation in poly(n) time.

- Example: $35 \rightarrow 5 \times 7$
- Fastest known classical algorithms: $2^{\tilde{O}(n^{1/3})}$ time (Pollard 1988; Buhler-Lenstra-Pomerance 1993)
- **Quantum algorithms:** poly(n) **time!** (Shor 1994)

Implication 1: Breaking Cryptography

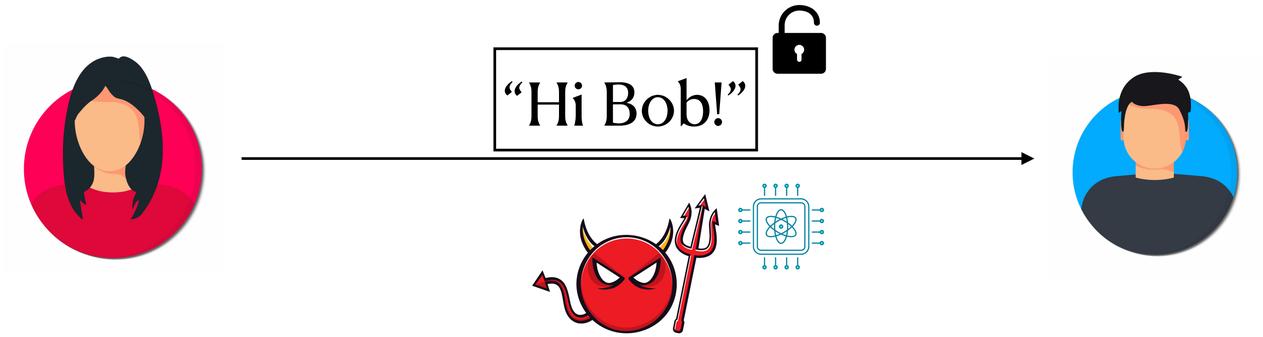
- RSA public-key cryptography:

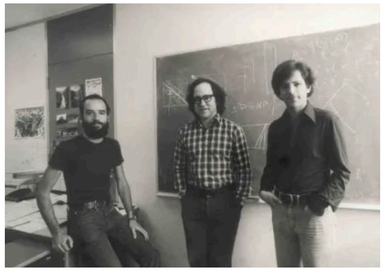




Implication 1: Breaking Cryptography

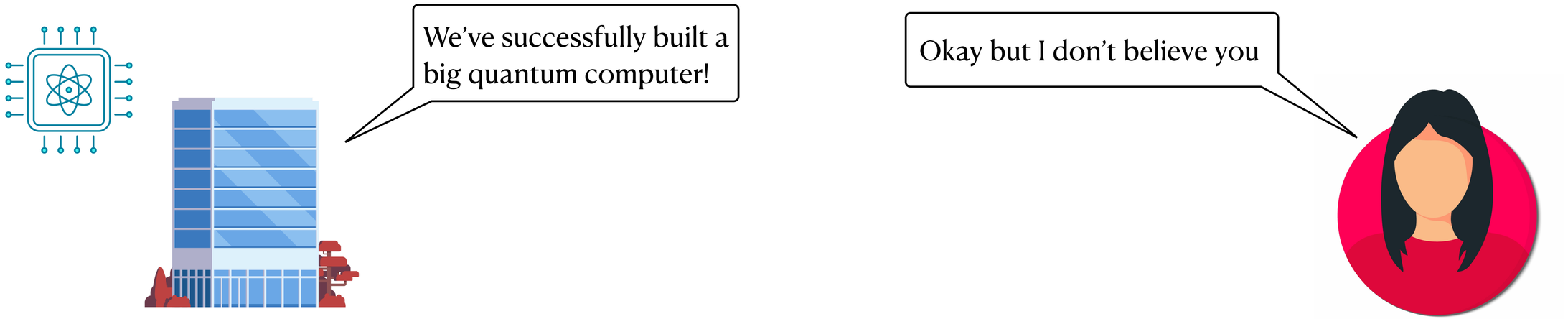
- RSA public-key cryptography:





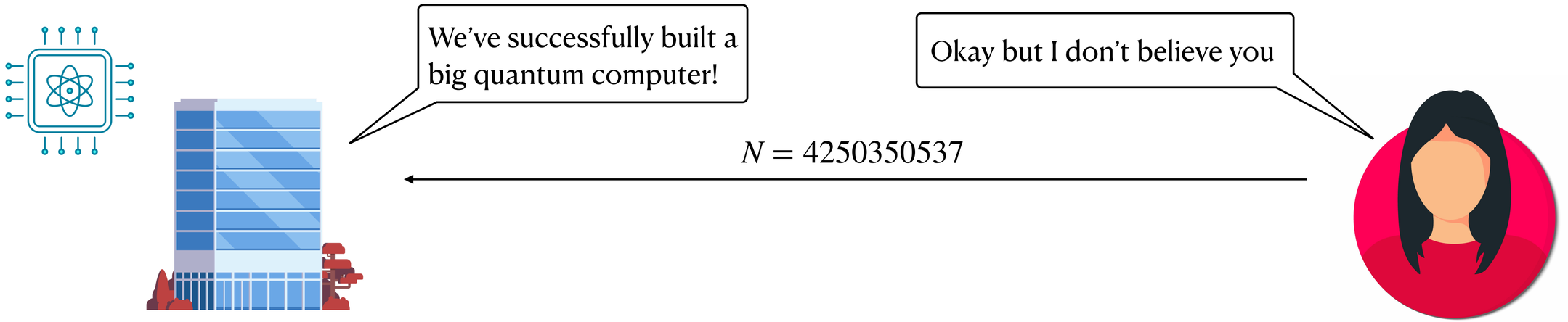
- **Completely broken if the eavesdropper has a large quantum computer (since then they could factor large integers)!**

Implication 2: Proofs of Quantumness



Q: How can XYZABC Labs convince Alice that they really do have a large quantum computer?

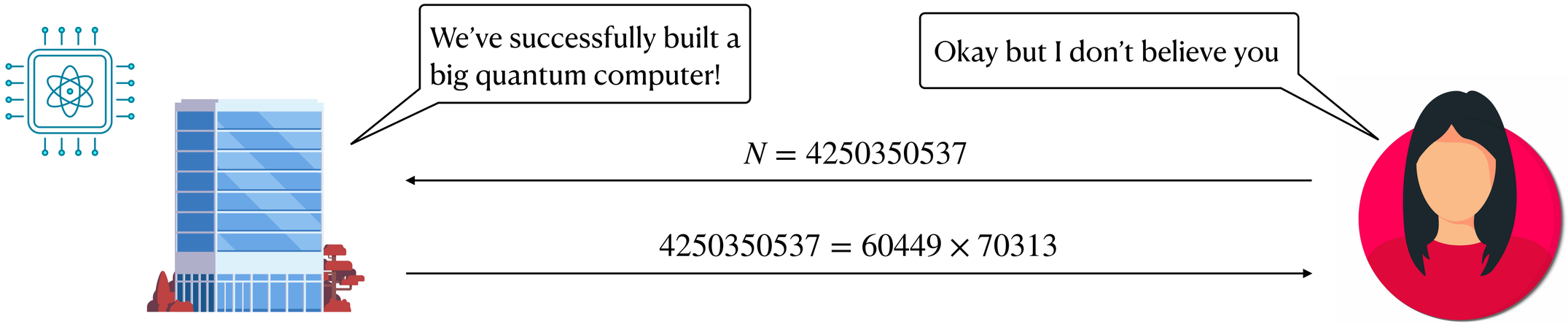
Implication 2: Proofs of Quantumness



Q: How can XYZABC Labs convince Alice that they really do have a large quantum computer?

One answer: By factoring a large integer of Alice's choice!

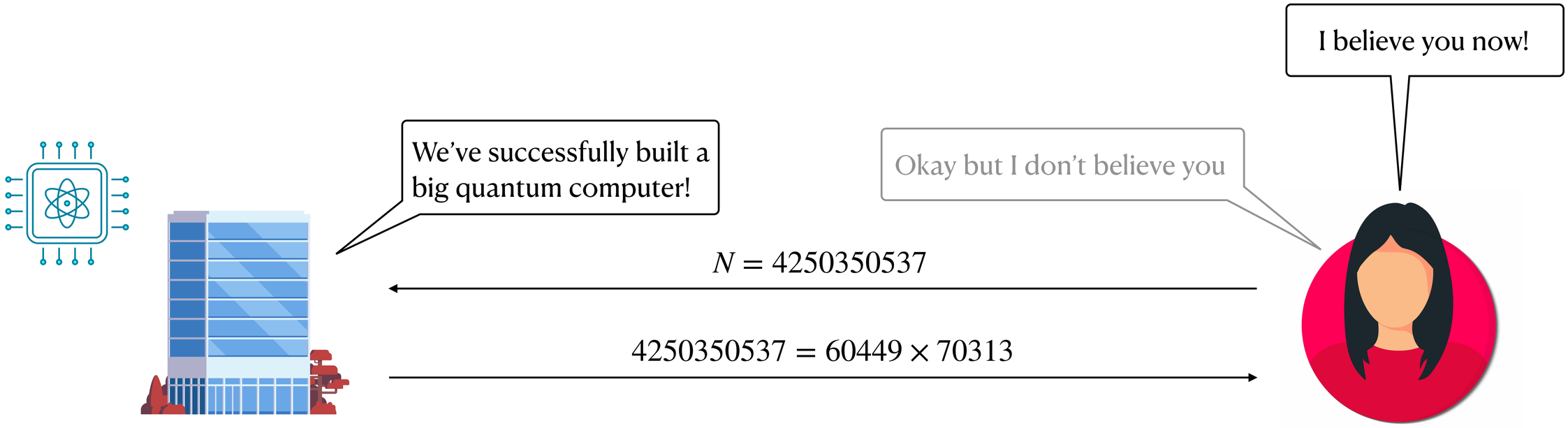
Implication 2: Proofs of Quantumness



Q: How can XYZABC Labs convince Alice that they really do have a large quantum computer?

One answer: By factoring a large integer of Alice's choice!

Implication 2: Proofs of Quantumness



Q: How can XYZABC Labs convince Alice that they really do have a large quantum computer?

One answer: By factoring a large integer of Alice's choice!

“Implication” 3: Intellectual Curiosity :)

- Useful toy problem for understanding the capability of quantum computing, in theory and practice
- Are there special types of integers that are especially easy to factor quantumly?
- How do these criteria compare with what makes an integer classically easy to factor?

“Implication” 3: Intellectual Curiosity :)

- Useful toy problem for understanding the capability of quantum computing, in theory and practice
- Are there special types of integers that are especially easy to factor quantumly?
- How do these criteria compare with what makes an integer classically easy to factor?

Coming up: some answers to these questions!

The Current State of Affairs

- Known for 30 years: if we had a large-scale quantum computer, we could factor large integers

The Current State of Affairs

- Known for 30 years: if we had a large-scale quantum computer, we could factor large integers
- **Slight catch: no-one has managed to build a large-scale quantum computer yet**

The Current State of Affairs

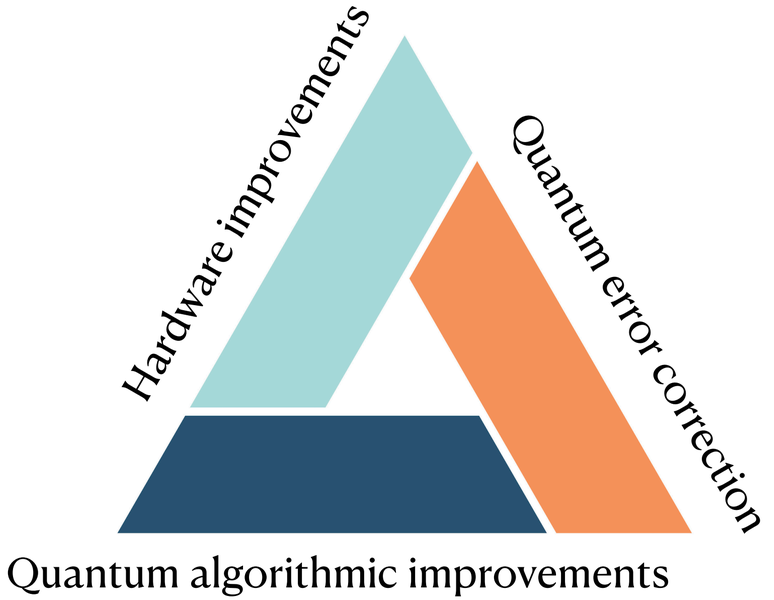
- Known for 30 years: if we had a large-scale quantum computer, we could factor large integers
- **Slight catch: no-one has managed to build a large-scale quantum computer yet**

"Our qubits are constantly trying to fall apart…
It's like you're trying to write in the sand and
the wind is blowing it away."

The Current State of Affairs

- Known for 30 years: if we had a large-scale quantum computer, we could factor large integers
- **Slight catch: no-one has managed to build a large-scale quantum computer yet**

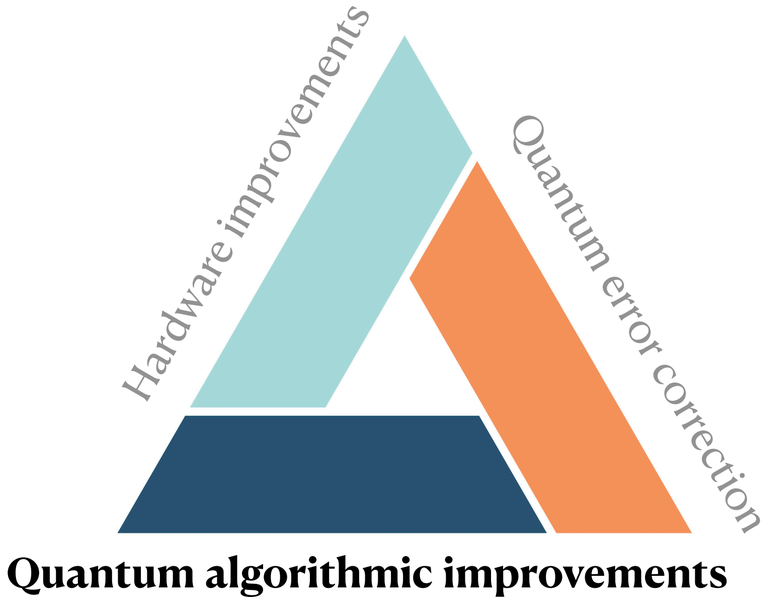
"Our qubits are constantly trying to fall apart…
It's like you're trying to write in the sand and
the wind is blowing it away."



The Current State of Affairs

- Known for 30 years: if we had a large-scale quantum computer, we could factor large integers
- **Slight catch: no-one has managed to build a large-scale quantum computer yet**

"Our qubits are constantly trying to fall apart…
It's like you're trying to write in the sand and
the wind is blowing it away."

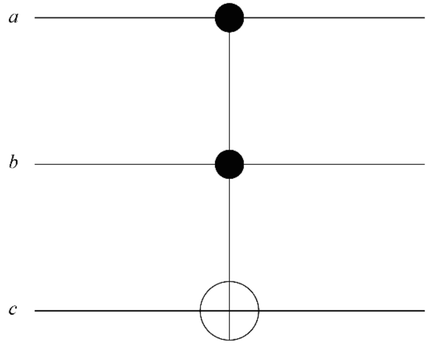


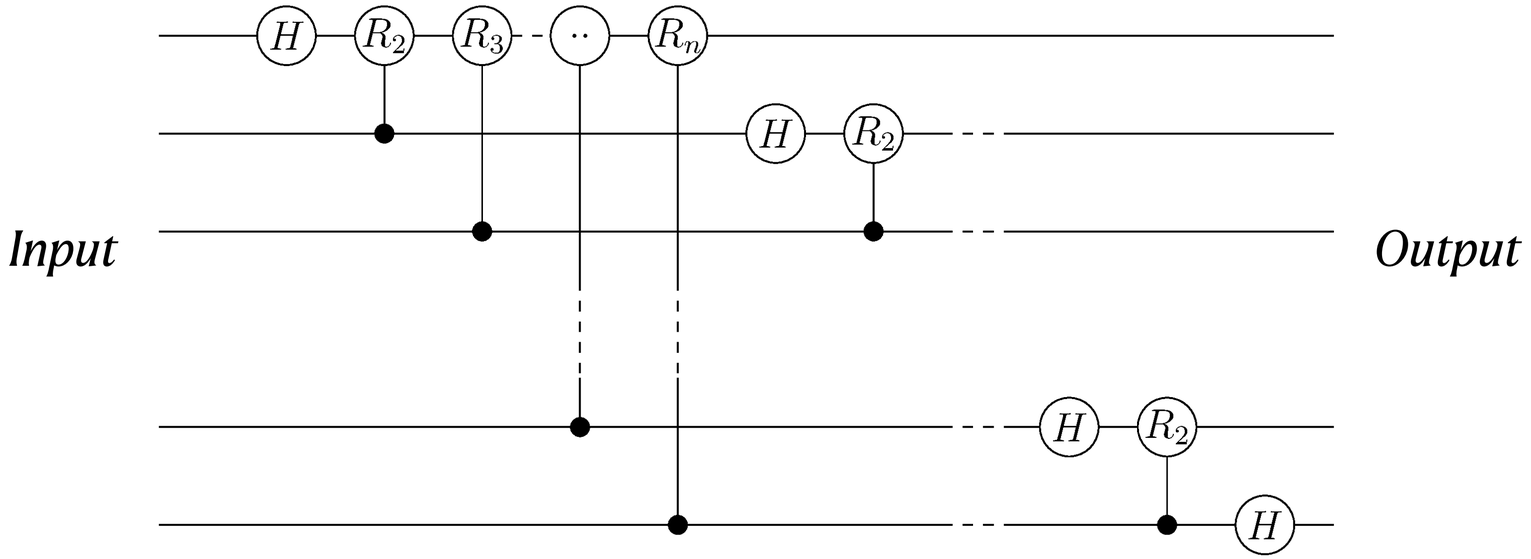
Efficiency Metrics for Quantum Circuits

- Like a classical circuit, but uses reversible logic gates on a fixed number of wires

Gates: the number of elementary operations (circles)

Space: the number of wires (lines)





Quantum Factoring Algorithms: A History

Authors	Types of inputs	Gates	Space
Shor (1994)	Any	$\tilde{O}(n^2)$	$\tilde{O}(n)$

n is the number of bits in the input N

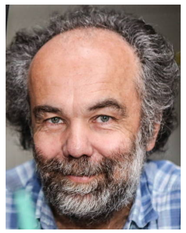
$\tilde{O}(\cdot)$ hides constant and $\text{poly}(\log n)$ factors

Quantum Factoring Algorithms: A History

Authors	Types of inputs	Gates	Space
Shor (1994)	Any	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Li-Peng-Du-Suter (2012)	$N = P^2Q$	$\tilde{O}(n)$	$\tilde{O}(n)$







Any input: would break RSA cryptography, and suffice as a proof of quantumness

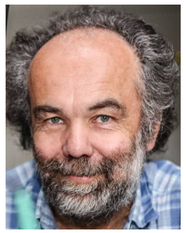
$N = P^2Q$: only suffices as a proof of quantumness

Quantum Factoring Algorithms: A History

Authors	Types of inputs	Gates	Space
Shor (1994)	Any	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Li-Peng-Du-Suter (2012)	$N = P^2Q$	$\tilde{O}(n)$	$\tilde{O}(n)$
Regev (2023)	Any	$\tilde{O}(n^{1.5})$	$O(n^{1.5})$





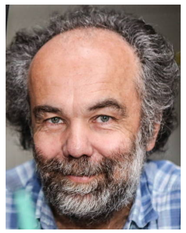



Quantum Factoring Algorithms: A History

Authors	Types of inputs	Gates	Space
Shor (1994)	Any	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Li-Peng-Du-Suter (2012)	$N = P^2Q$	$\tilde{O}(n)$	$\tilde{O}(n)$
Regev (2023)	Any	$\tilde{O}(n^{1.5})$	$O(n^{1.5})$
R-Vaikuntanathan (2024)	Any	$\tilde{O}(n^{1.5})$	$\tilde{O}(n)$





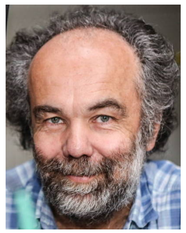



Quantum Factoring Algorithms: A History

Authors	Types of inputs	Gates	Space
Shor (1994)	Any	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Li-Peng-Du-Suter (2012)	$N = P^2Q$	$\tilde{O}(n)$	$\tilde{O}(n)$
Regev (2023)	Any	$\tilde{O}(n^{1.5})$	$O(n^{1.5})$
R-Vaikuntanathan (2024)	Any	$\tilde{O}(n^{1.5})$	$\tilde{O}(n)$
K**R**VV (2024)	$N = P^2Q$ $(Q \ll P)$	$\tilde{O}(n)$	$\tilde{O}(n^{0.67})$







Factoring: What Next?

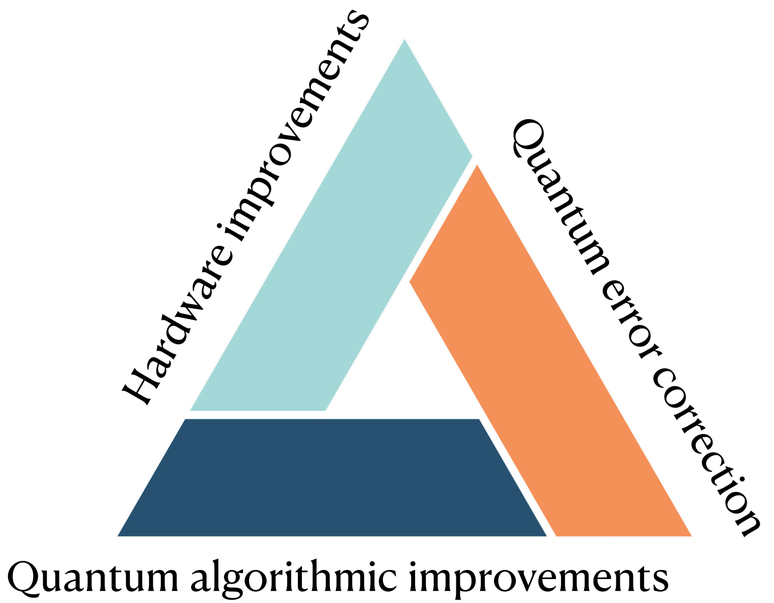
- Ultimate goal: factor a large integer on a real quantum computer

Factoring: What Next?

- Ultimate goal: factor a large integer on a real quantum computer
- Algorithmically: nail down and optimise the **concrete** costs of these algorithms

Factoring: What Next?

- Ultimate goal: factor a large integer on a real quantum computer
- Algorithmically: nail down and optimise the **concrete** costs of these algorithms
- Interdisciplinary collaboration with hardware, error correction researchers needed for further improvements



Non-Interactive Q+A

- *Q1: Why factoring, as opposed to elliptic curve discrete log (ECDLP)? ECDLP is classically harder, and Shor's algorithm also solves this → we can work with smaller keys*

Non-Interactive Q+A

- *Q1: Why factoring, as opposed to elliptic curve discrete log (ECDLP)? ECDLP is classically harder, and Shor's algorithm also solves this → we can work with smaller keys*
  - A1: Elliptic curve arithmetic seems painful to carry out on a quantum computer in practice

Non-Interactive Q+A

- *Q1: Why factoring, as opposed to elliptic curve discrete log (ECDLP)? ECDLP is classically harder, and Shor's algorithm also solves this → we can work with smaller keys*
  - A1: Elliptic curve arithmetic seems painful to carry out on a quantum computer in practice
  - A2: Factoring seems to admit a richer variety of quantum algorithms

Non-Interactive Q+A

- *Q1: Why factoring, as opposed to elliptic curve discrete log (ECDLP)? ECDLP is classically harder, and Shor's algorithm also solves this → we can work with smaller keys*
  - A1: Elliptic curve arithmetic seems painful to carry out on a quantum computer in practice
  - A2: Factoring seems to admit a richer variety of quantum algorithms
- *Q2: Are we really going to see quantum computers on the scale necessary to factor large integers?*

Non-Interactive Q+A

- *Q1: Why factoring, as opposed to elliptic curve discrete log (ECDLP)? ECDLP is classically harder, and Shor's algorithm also solves this → we can work with smaller keys*
  - A1: Elliptic curve arithmetic seems painful to carry out on a quantum computer in practice
  - A2: Factoring seems to admit a richer variety of quantum algorithms
- *Q2: Are we really going to see quantum computers on the scale necessary to factor large integers?*

$2040 says we'll be able to quantumly factor 2040-bit integers by 2040

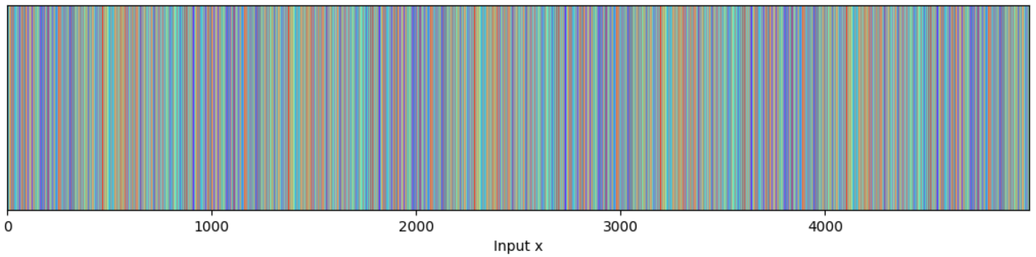
Part 2: Proofs of Quantumness from Factoring P^2Q

Gregory D. Kahanamoku-Meyer*, Seyoon Ragavan*, Vinod Vaikuntanathan*, Katherine Van Kirk†


*MIT, †Harvard

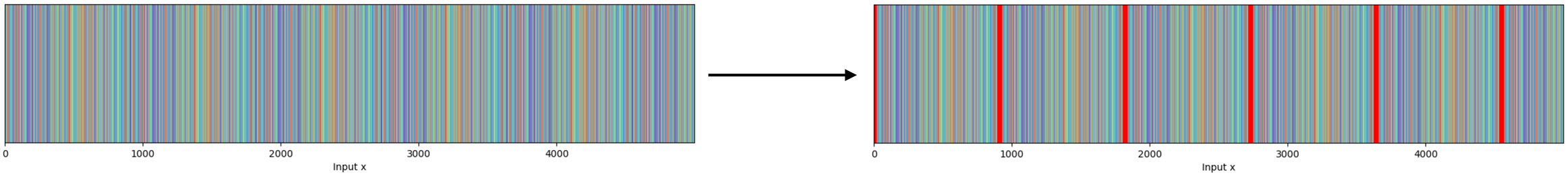

Preliminary: Quantum Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown period T
  - $x \equiv y \pmod{T} \Leftrightarrow f(x) = f(y)$



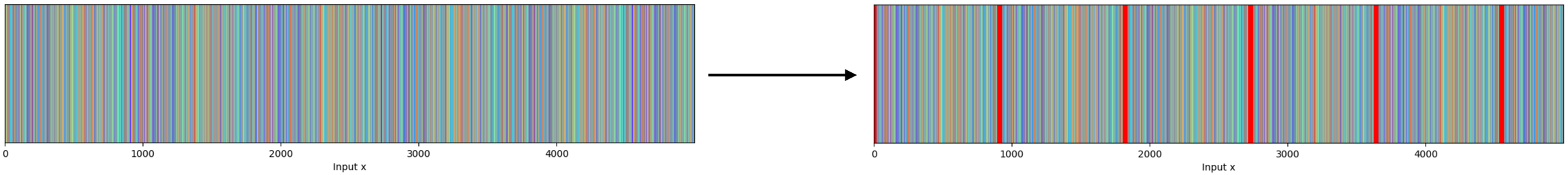
Preliminary: Quantum Period Finding

- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown period T
  - $x \equiv y \pmod{T} \Leftrightarrow f(x) = f(y)$
- Informal theorem statement: can quantumly recover T using essentially only the gates/space needed to compute $f(x)$ for $|x| \leq \text{poly}(T)$



Preliminary: Quantum Period Finding

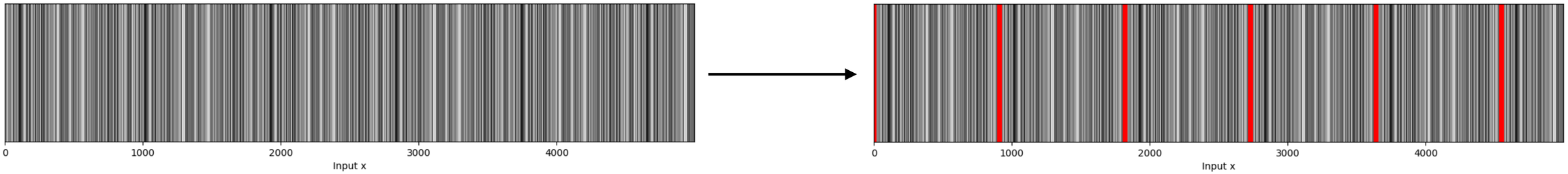
- Strictly periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown period T
  - $x \equiv y \pmod{T} \Leftrightarrow f(x) = f(y)$
- Informal theorem statement: can quantumly recover T using essentially only the gates/space needed to compute $f(x)$ for $|x| \leq \text{poly}(T)$
- Also the centrepiece of Shor's quantum factoring algorithm



Preliminary: General Quantum Period Finding

May and Schlieper (2022)

- ~~Strictly~~ periodic function $f : \mathbb{Z} \to \mathbb{Z}$ with unknown period T
  - $x \equiv y \pmod{T} \Rightarrow f(x) = f(y)$
- Informal theorem* statement: can quantumly recover T using essentially only the gates/space needed to compute $f(x)$ for $|x| \leq \text{poly}(T)$
- In fact, the algorithm is exactly the same!



* modulo some technical caveats

Preliminary: The Legendre Symbol

- a is a *quadratic residue* modulo an odd prime p if there exists integer x such that $a \equiv x^2 \pmod{p}$

Preliminary: The Legendre Symbol

- a is a *quadratic residue* modulo an odd prime p if there exists integer x such that $a \equiv x^2 \pmod{p}$
- Legendre symbol essentially indicates whether this is the case:

$$\left(\frac{a}{p}\right) = \begin{cases} 1, \text{ if } a \text{ is a nonzero quadratic residue modulo } p; \text{ and} \\ -1, \text{ if } a \text{ is not a quadratic residue modulo } p; \text{ and} \\ 0, \text{ if } a \text{ divisible by } p\,. \end{cases}$$

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli
- For $N = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$, define:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_r}\right)^{\alpha_r}$$

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli
- For $N = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$, define:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_r}\right)^{\alpha_r}$$

- Note for intuition: the quadratic residue characterisation does **not** carry over from the Legendre symbol

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli
- For $N = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$, define:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_r}\right)^{\alpha_r}$$

- Note for intuition: the quadratic residue characterisation does **not** carry over from the Legendre symbol
  - Could have $\left(\frac{a}{N}\right) = 1$ without a being a quadratic residue modulo N

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli
- For $N = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$, define:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_r}\right)^{\alpha_r}$$

- Useful property: $a \equiv b \pmod{N} \Rightarrow \left(\frac{a}{N}\right) = \left(\frac{b}{N}\right)$

Preliminary: The Jacobi Symbol

- Essentially generalises the Legendre symbol to odd composite moduli
- For $N = p_1^{\alpha_1} \ldots p_r^{\alpha_r}$, define:

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \ldots \left(\frac{a}{p_r}\right)^{\alpha_r}$$

- Useful property: $a \equiv b \pmod{N} \Rightarrow \left(\frac{a}{N}\right) = \left(\frac{b}{N}\right)$
- Theorem (Schönhage 1971): can compute $\left(\frac{a}{N}\right)$ in time $\tilde{O}(\log N)$

Quantumly Factoring $N = P^2 Q$

Li, Peng, Du, Suter (2012)

- Simple observation:*

$$\left(\frac{a}{N}\right) = \left(\frac{a}{P}\right)^2 \left(\frac{a}{Q}\right) = \left(\frac{a}{Q}\right), \text{ which is periodic with period } Q!$$

* modulo minor technical caveats; could have $\left(\frac{a}{P}\right) = 0$

Quantumly Factoring $N = P^2 Q$

Li, Peng, Du, Suter (2012)

- Simple observation:*

$$\left(\frac{a}{N}\right) = \left(\frac{a}{P}\right)^2 \left(\frac{a}{Q}\right) = \left(\frac{a}{Q}\right), \text{ which is periodic with period } Q!$$

- So quantum period finding → recover Q (and hence P)

* modulo minor technical caveats; could have $\left(\frac{a}{P}\right) = 0$

Quantumly Factoring $N = P^2Q$

Li, Peng, Du, Suter (2012)

- Simple observation:*

$$\left(\frac{a}{N}\right) = \left(\frac{a}{P}\right)^2 \left(\frac{a}{Q}\right) = \left(\frac{a}{Q}\right), \text{ which is periodic with period } Q!$$

- So quantum period finding → recover Q (and hence P)
- Gate complexity: cost of computing $\left(\frac{a}{N}\right)$ for $a \leq \mathsf{poly}(Q)$, which is $\tilde{O}(\log N)$

* modulo minor technical caveats; could have $\left(\frac{a}{P}\right) = 0$

Quantumly Factoring $N = P^2Q$

Li, Peng, Du, Suter (2012)

- Simple observation:*

$$\left(\frac{a}{N}\right) = \left(\frac{a}{P}\right)^2 \left(\frac{a}{Q}\right) = \left(\frac{a}{Q}\right), \text{ which is periodic with period } Q!$$

- So quantum period finding → recover Q (and hence P)
- Gate complexity: cost of computing $\left(\frac{a}{N}\right)$ for $a \leq \mathsf{poly}(Q)$, which is $\tilde{O}(\log N)$
- Space (if naively implemented): also $\tilde{O}(\log N)$

* modulo minor technical caveats; could have $\left(\frac{a}{P}\right) = 0$

Pushing Space Down to $\tilde{O}(\log Q)$!

Kahanamoku-Meyer, R, Vaikuntanathan, Van Kirk (2024)

- A new way to compute $\left(\frac{a}{N}\right)$, exploiting two observations:
  - $\log a = O(\log Q)$ could potentially be much less than $O(\log N)$

Pushing Space Down to $\tilde{O}(\log Q)$!

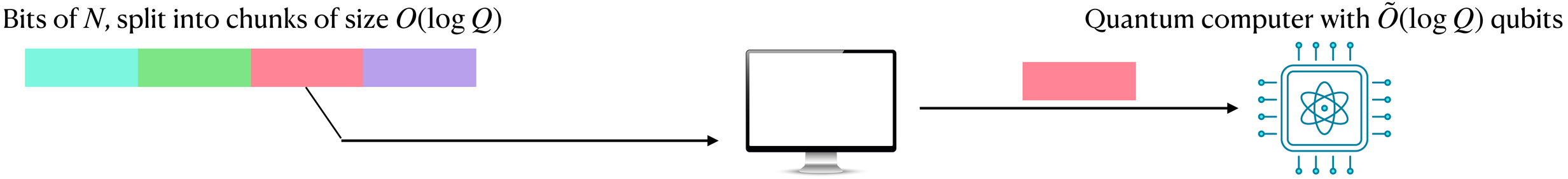
Kahanamoku-Meyer, R, Vaikuntanathan, Van Kirk (2024)

- A new way to compute $\left(\frac{a}{N}\right)$, exploiting two observations:
  - $\log a = O(\log Q)$ could potentially be much less than $O(\log N)$
  - N is classically known → could quantumly "stream" through bits of N to save space

Pushing Space Down to $\tilde{O}(\log Q)$!

Kahanamoku-Meyer, R, Vaikuntanathan, Van Kirk (2024)

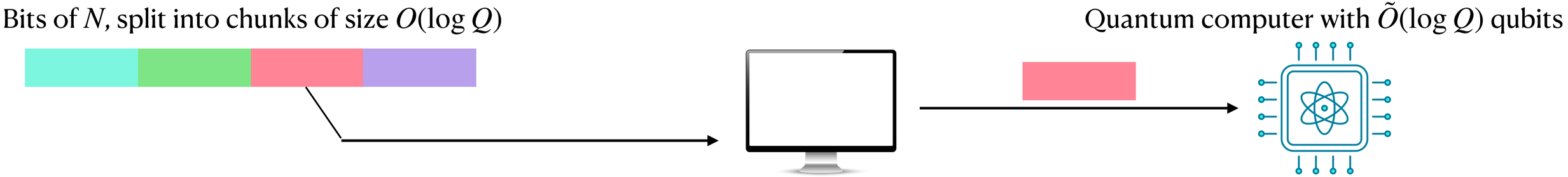
- A new way to compute $\left(\frac{a}{N}\right)$, exploiting two observations:
  - $\log a = O(\log Q)$ could potentially be much less than $O(\log N)$
  - N is classically known → could quantumly "stream" through bits of N to save space



Pushing Space Down to $\tilde{O}(\log Q)$!

Kahanamoku-Meyer, R, Vaikuntanathan, Van Kirk (2024)

- A new way to compute $\left(\frac{a}{N}\right)$, exploiting two observations:
  - $\log a = O(\log Q)$ could potentially be much less than $O(\log N)$
  - N is classically known → could quantumly "stream" through bits of N to save space
- Resulting circuit: gates still $\tilde{O}(\log N)$ and space is now $\tilde{O}(\log Q)$!



Setting Parameters

- Balancing act:
  - If Q is too large: our quantum circuit is no better than the LPDS12 circuit

Setting Parameters

- Balancing act:
  - If Q is too large: our quantum circuit is no better than the LPDS12 circuit
  - If Q is too small: classical algorithms could exploit this structure to run faster than general-purpose classical factoring algorithms

Setting Parameters

- Balancing act:
  - If Q is too large: our quantum circuit is no better than the LPDS12 circuit
  - If Q is too small: classical algorithms could exploit this structure to run faster than general-purpose classical factoring algorithms
- Sweet spot: $\log Q = \tilde{O}((\log N)^{2/3}) \rightarrow$ space $\tilde{O}((\log N)^{2/3})$, gates $\tilde{O}(\log N)$

More Open Questions

- Other quantum factoring algorithms exploiting special structure in N? (Many such algorithms in the classical world)

More Open Questions

- Other quantum factoring algorithms exploiting special structure in N? (Many such algorithms in the classical world)
- Can these algorithms be boosted (e.g. using number theory magic) to factor generic integers N, or at least RSA integers $N = PQ$, more efficiently?

More Open Questions

- Other quantum factoring algorithms exploiting special structure in N? (Many such algorithms in the classical world)
- Can these algorithms be boosted (e.g. using number theory magic) to factor generic integers N, or at least RSA integers $N = PQ$, more efficiently?
  - Current generalisation: can **completely** factor $N = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$ for distinct $\alpha_1, \ldots, \alpha_r$