Workshop: quantum algorithms for asymmetric cryptography

Greg Meyer [1]

[1]MIT

July 9, 2026

Most important open problem: why we are all here

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)
- Quantum-vulnerable crypto? (6/10, I guess…)

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)
- Quantum-vulnerable crypto? (6/10, I guess…)
- …?

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)
- Quantum-vulnerable crypto? (6/10, I guess…)
- …?

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)
- Quantum-vulnerable crypto? (6/10, I guess…)
- …?

Let me know if you have ideas; we can vote on Friday.

Most important open problem: why we are all here

There is no standard name for not-post-quantum crypto

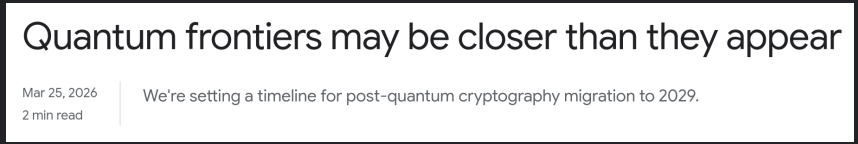
What should we call it?

- Pre-quantum crypto? (5/10, sounds weird…)
- “Classical” cryptography? (2/10, gives the wrong idea…)
- Quantum-vulnerable crypto? (6/10, I guess…)
- …?

Let me know if you have ideas; we can vote on Friday.
We’re on a timeline to figure out a name because it will be dead soon…

Why we are actually here: quantum attacks are coming

Google:



Cloudflare:

Why we are actually here: quantum attacks are coming

US government:

Why we are actually here: quantum attacks are coming

As experts in those quantum attacks, what do we do about it?

Roadmap

Goals of the workshop

Roadmap

Goals of the workshop

Landscape of vulnerable cryptography

Roadmap

Goals of the workshop

Landscape of vulnerable cryptography

Some history: how did we get here?

Roadmap

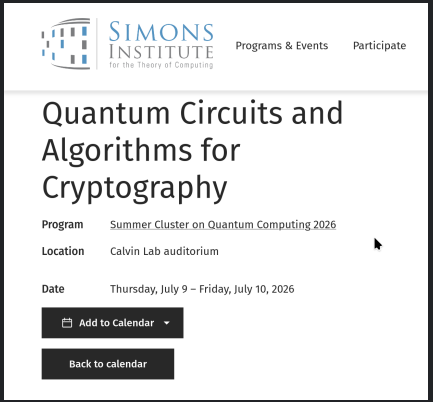
Goals of the workshop

Landscape of vulnerable cryptography

Some history: how did we get here?

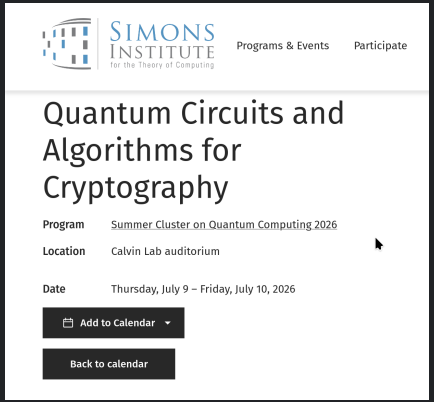
Workshop goals

Discuss the *operational* status of quantum cryptanalysis



Workshop goals

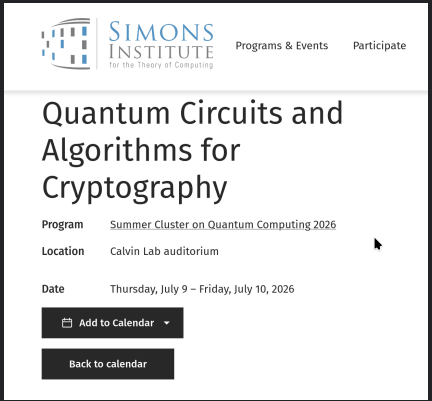
Discuss the *operational* status of quantum cryptanalysis



- Focus on quantum algorithms and logical circuits

Workshop goals

Discuss the *operational* status of quantum cryptanalysis



- Focus on quantum algorithms and logical circuits
- What *ideas/tricks* are important for the currently best circuits?

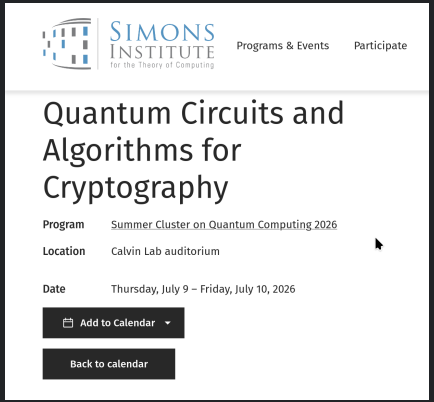
Workshop goals

Discuss the *operational* status of quantum cryptanalysis

- Focus on quantum algorithms and logical circuits
- What *ideas/tricks* are important for the currently best circuits?
- What ideas/tricks might become important?

Workshop goals

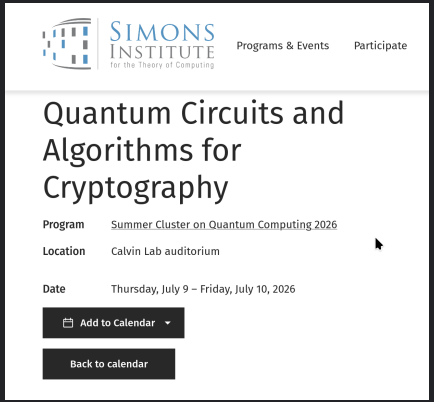
Discuss the *operational* status of quantum cryptanalysis



- Focus on quantum algorithms and logical circuits
- What *ideas/tricks* are important for the currently best circuits?
- What ideas/tricks might become important?
- Where might further progress lie?

Workshop goals

Discuss the *operational* status of quantum cryptanalysis



- Focus on quantum algorithms and logical circuits
- What *ideas/tricks* are important for the currently best circuits?
- What ideas/tricks might become important?
- Where might further progress lie?

Workshop goals

Discuss the *operational* status of quantum cryptanalysis

- Focus on quantum algorithms and logical circuits
- What *ideas/tricks* are important for the currently best circuits?
- What ideas/tricks might become important?
- Where might further progress lie?

Don't be afraid to interrupt with questions, ideas, etc.!

Workshop schedule

	Thursday	Friday
9:30-10:30	Speaker: Greg Meyer	Speaker: Craig Gidney
11:00-12:00	Speaker: Martin Ekerå	Speaker: André Schrottenloher
2:00-3:00	Speaker: Seyoon Ragavan	Panel 2: *Responsible disclosure*
3:30-5:00	Panel 1: *Open directions*	Lightning talks/discussion

Panel 1: Open directions

- Martin Ekerå (Swedish NCSA and KTH)
- Craig Gidney (Google)
- André Schrottenloher (INRIA)

Panel 2: Responsible disclosure

- Ryan Babbush (Google)
- Rolfe Schmidt (Signal)
- Martin Ekerå (Swedish NCSA and KTH)
- André Schrottenloher (INRIA)

Workshop schedule

	Thursday	Friday
9:30-10:30	Speaker: Greg Meyer	Speaker: Craig Gidney
11:00-12:00	Speaker: Martin Ekerå	Speaker: André Schrottenloher
2:00-3:00	Speaker: Seyoon Ragavan	Panel 2: *Responsible disclosure*
3:30-5:00	Panel 1: *Open directions*	Lightning talks/discussion

Lightning talks:

- A 5-10 minute talk (whiteboard or otherwise)

Workshop schedule

	Thursday	Friday
9:30-10:30	Speaker: Greg Meyer	Speaker: Craig Gidney
11:00-12:00	Speaker: Martin Ekerå	Speaker: André Schrottenloher
2:00-3:00	Speaker: Seyoon Ragavan	Panel 2: *Responsible disclosure*
3:30-5:00	Panel 1: *Open directions*	Lightning talks/discussion

Lightning talks:

- A 5-10 minute talk (whiteboard or otherwise)
- Not recorded

Workshop schedule

	Thursday	Friday
9:30-10:30	Speaker: Greg Meyer	Speaker: Craig Gidney
11:00-12:00	Speaker: Martin Ekerå	Speaker: André Schrottenloher
2:00-3:00	Speaker: Seyoon Ragavan	Panel 2: *Responsible disclosure*
3:30-5:00	Panel 1: *Open directions*	Lightning talks/discussion

Lightning talks:

- A 5-10 minute talk (whiteboard or otherwise)
- Not recorded
- Email me <gkm@mit.edu> (or just tell me) if you want to speak!

Workshop schedule

	Thursday	Friday
9:30-10:30	Speaker: Greg Meyer	Speaker: Craig Gidney
11:00-12:00	Speaker: Martin Ekerå	Speaker: André Schrottenloher
2:00-3:00	Speaker: Seyoon Ragavan	Panel 2: *Responsible disclosure*
3:30-5:00	Panel 1: *Open directions*	Lightning talks/discussion

Lightning talks:

- A 5-10 minute talk (whiteboard or otherwise)
- Not recorded
- Email me <gkm@mit.edu> (or just tell me) if you want to speak!
- Impromptu presentations also OK

Roadmap

Goals of the workshop

Landscape of vulnerable cryptography

Some history: how did we get here?

Abstract computational problems

Discrete logarithm

Consider a multiplicative group $\mathbb{G}$ with generator g.

Given a group element $x \in \mathbb{G}$, **find d such that $g^d = x$.**

Integer factorization

Consider an integer N.

Find its prime decomposition.

Vulnerable concrete problems used in cryptography

Elliptic curve discrete log
(ECDLP)

For an elliptic curve group $\mathbb{G}$, consider two curve points g and $x = [d]g$.

Find d.

Vulnerable concrete problems used in cryptography

Elliptic curve discrete log
(ECDLP)

For an elliptic curve group $\mathbb{G}$, consider two curve points g and $x = [d]g$.

Find d.

Finite field discrete log
(FFDLP)

Consider an integer p (typically a prime). Given positive integers g and $x \equiv g^d \pmod{p}$:

Find d.

Vulnerable concrete problems used in cryptography

Elliptic curve discrete log
(ECDLP)

For an elliptic curve group $\mathbb{G}$, consider two curve points g and $x = [d]g$.

Find d.

Finite field discrete log
(FFDLP)

Consider an integer p (typically a prime). Given positive integers g and $x \equiv g^d \pmod{p}$:

Find d.

RSA integer factorization
(IFP)

Consider an integer $N = pq$, where p, q are primes of equal bit length.

Find p **and** q**.**

Vulnerable concrete problems used in cryptography

Elliptic curve discrete log
(ECDLP)

For an elliptic curve group $\mathbb{G}$, consider two curve points g and $x = [d]g$.

Find d.

Finite field discrete log
(FFDLP)

Consider an integer p (typically a prime). Given positive integers g and $x \equiv g^d \pmod{p}$:

Find d.

RSA integer factorization
(IFP)

Consider an integer $N = pq$, where p, q are primes of equal bit length.

Find p **and** q**.**

Note: cryptosystems rely on slightly stronger hardness assumptions

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

	Attention given	How widely used?	Quantum cost?
1.	IFP		

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?
1. IFP		
2. ECDLP		

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?
1. IFP		
2. ECDLP		
3. FFDLP		

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?
1. IFP	1. ECDLP	
2. ECDLP		
3. FFDLP		

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?
1. IFP	1. ECDLP	
2. ECDLP	2. IFP	
3. FFDLP		

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost?
1. IFP	1. ECDLP	
2. ECDLP	2. IFP	
3. FFDLP	3. FFDLP	

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost? (1 = most expensive)
1. IFP	1. ECDLP	
2. ECDLP	2. IFP	
3. FFDLP	3. FFDLP	

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given

1. IFP
2. ECDLP
3. FFDLP

How widely used?

1. ECDLP
2. IFP
3. FFDLP

Quantum cost?

1 = most expensive

1. IFP

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given

1. IFP
2. ECDLP
3. FFDLP

How widely used?

1. ECDLP
2. IFP
3. FFDLP

Quantum cost?

1 = most expensive

1. IFP
2. ECDLP

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given

1. IFP
2. ECDLP
3. FFDLP

How widely used?

1. ECDLP
2. IFP
3. FFDLP

Quantum cost?

1 = most expensive

1. IFP
2. ECDLP
3. FFDLP

Comparing the vulnerable problems

Note: this entire slide is my qualitative assessment, in July 2026

IFP = integer factorization; ECDLP = elliptic curve discrete log; FFDLP = finite field discrete log

Attention given	How widely used?	Quantum cost? (1 = most expensive)
1. IFP	1. ECDLP	1. IFP
2. ECDLP	2. IFP	2. ECDLP
3. FFDLP	3. FFDLP	3. FFDLP

Takeaway: factoring gets by far the *most attention*, but not seemingly for any *good reason*.

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$$n = \lceil \log_2 |\mathbb{G}| \rceil$$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$$n = \lceil \log_2 |\mathbb{G}| \rceil$$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$$n = \lceil \log_2 |\mathbb{G}| \rceil$$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

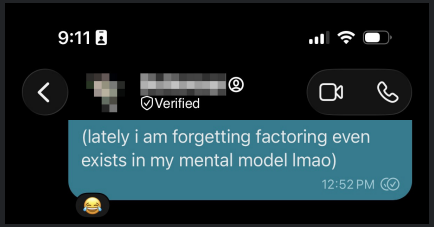
Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

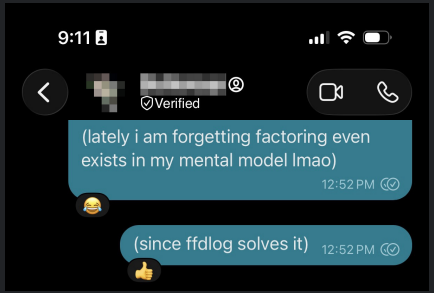
- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

…

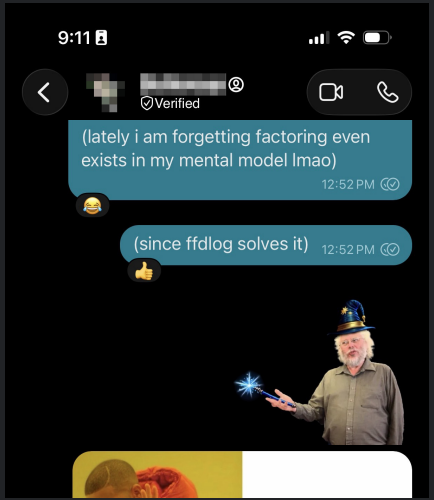
A window into my scientific process



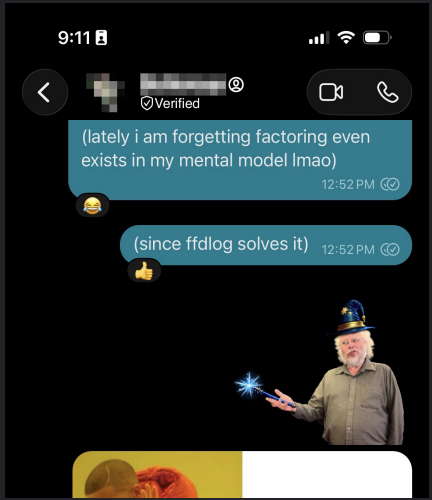
A window into my scientific process

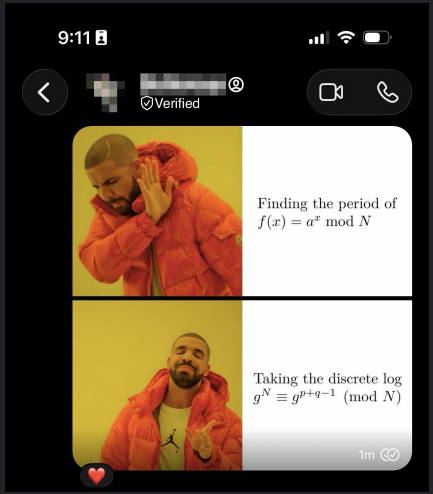


A window into my scientific process



A window into my scientific process





Why discrete log is the thing to focus on

Factoring RSA numbers reduces to discrete log;

Why discrete log is the thing to focus on

Factoring RSA numbers reduces to discrete log;
furthermore, doing it this way is actually **more efficient**.

Why discrete log is the thing to focus on

Factoring RSA numbers reduces to discrete log;
furthermore, doing it this way is actually **more efficient**.

→ You will hear about this reduction (and more) from Martin
during his **11 AM-12 PM talk today**!

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

RSA integer factorization
(IFP)

Given $N = pq$, find p and q.

$n = \lceil \log_2 N \rceil$

$\Leftarrow$ FFDLP with params.
$n = \lceil \log_2 N \rceil$, $m = n/2$

Quantum cost: rough overview

Elliptic curve discrete log
(ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log
(FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

RSA integer factorization
(IFP)

Given $N = pq$, find p and q.

$n = \lceil \log_2 N \rceil$

$\Leftarrow$ FFDLP with params.
$n = \lceil \log_2 N \rceil$, $m = n/2$

For $n = 2048$:

- Qubits: ~ 1400
- Toffolis: $\sim 2^{33}$

Quantum cost: rough overview

Elliptic curve discrete log (ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log (FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

RSA integer factorization (IFP)

Given $N = pq$, find p and q.

$n = \lceil \log_2 N \rceil$

$\Leftarrow$ FFDLP with params.
$n = \lceil \log_2 N \rceil$, $m = n/2$

For $n = 2048$:

- Qubits: ~ 1400
- Toffolis: $\sim 2^{33}$

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*?

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Metrics for cost

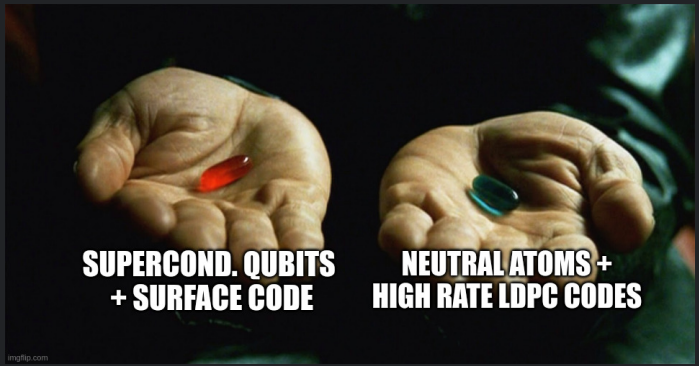
How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!



Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code
+ superconducting

High rate LDPC
+ neutral atoms

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code
+ superconducting

- Fast cycle time

High rate LDPC
+ neutral atoms

- Slow cycle time

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code
+ superconducting

- Fast cycle time
- Many physical qubits per logical

High rate LDPC
+ neutral atoms

- Slow cycle time
- Fewer physical qubits per logical

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code
+ superconducting

- Fast cycle time
- Many physical qubits per logical
- Parallelism has big space footprint

High rate LDPC
+ neutral atoms

- Slow cycle time
- Fewer physical qubits per logical
- Parallelism more achievable

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code + superconducting

- Fast cycle time
- Many physical qubits per logical
- Parallelism has big space footprint
- **Minimize logical qubits at (almost) all costs**

High rate LDPC + neutral atoms

- Slow cycle time
- Fewer physical qubits per logical
- Parallelism more achievable

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code
+ superconducting

- Fast cycle time
- Many physical qubits per logical
- Parallelism has big space footprint
- **Minimize logical qubits at (almost) all costs**

High rate LDPC
+ neutral atoms

- Slow cycle time
- Fewer physical qubits per logical
- Parallelism more achievable
- **Balance depth and logical qubit count?**

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code + superconducting

- Fast cycle time
- Many physical qubits per logical
- Parallelism has big space footprint
- **Minimize logical qubits at (almost) all costs**

High rate LDPC + neutral atoms

- Slow cycle time
- Fewer physical qubits per logical
- Parallelism more achievable
- **Balance depth and logical qubit count?**

Many algorithmic decisions made with **surface code in mind.**

Metrics for cost

How should we balance logical *qubits* vs *Toffolis*? (not to mention depth, other gates, ...)

Depends strongly on architecture!

Surface code + superconducting

- Fast cycle time
- Many physical qubits per logical
- Parallelism has big space footprint
- **Minimize logical qubits at (almost) all costs**

High rate LDPC + neutral atoms

- Slow cycle time
- Fewer physical qubits per logical
- Parallelism more achievable
- **Balance depth and logical qubit count?**

Many algorithmic decisions made with **surface code in mind.**
Take a second look at old constructions under new heuristics?

Quantum cost: rough overview

Elliptic curve discrete log (ECDLP)

Given $x = [d]g$, find d.

$n = \lceil \log_2 |\mathbb{G}| \rceil$

Qubits: $\sim 4.4n + O(\sqrt{n})$
Gates: $\widetilde{O}(n^3)$

For $n = 256$:

- Qubits: ~ 1200
- Toffolis: $\sim 2^{26}$

Finite field discrete log (FFDLP)

Given $x \equiv g^d \pmod{p}$ find d.

$n = \lceil \log p \rceil \quad m = \lceil \log d \rceil$

Qubits: $m + o(m) + O(\log n)$
Gates: $\widetilde{O}(mn^2)$

For $n = 2048$, $m = 225$:

- Qubits: ~ 350
- Toffolis: $\sim 2^{26}$

RSA integer factorization (IFP)

Given $N = pq$, find p and q.

$n = \lceil \log_2 N \rceil$

$\Leftarrow$ FFDLP with params.
$n = \lceil \log_2 N \rceil$, $m = n/2$

For $n = 2048$:

- Qubits: ~ 1400
- Toffolis: $\sim 2^{33}$

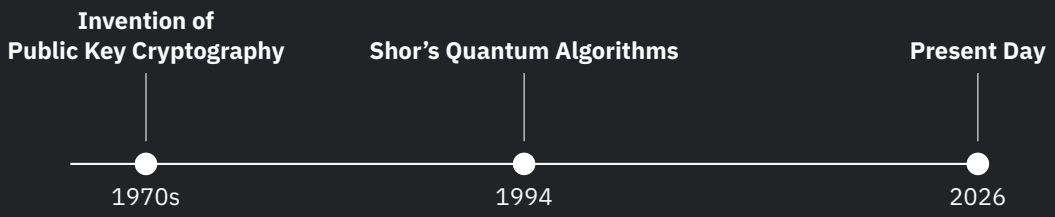
Roadmap

Goals of the workshop

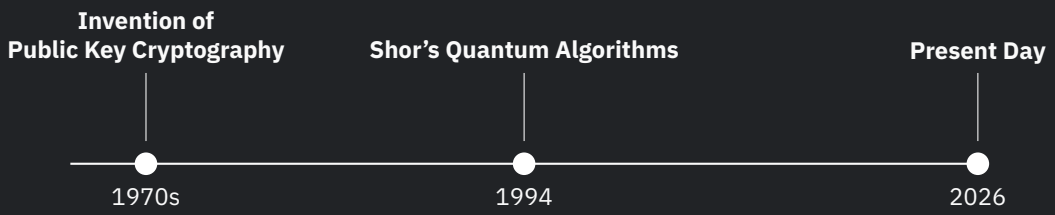
Landscape of vulnerable cryptography

Some history: how did we get here?

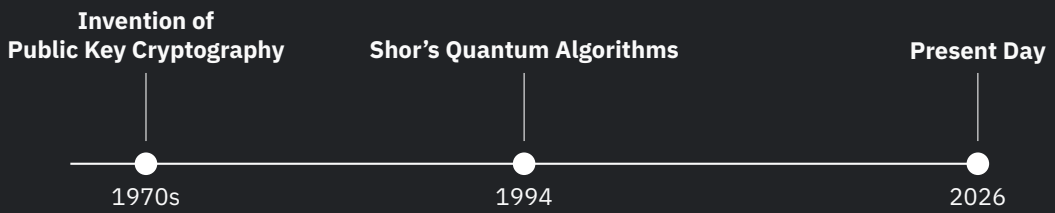
Timeline of public-key cryptography



Timeline of public-key cryptography

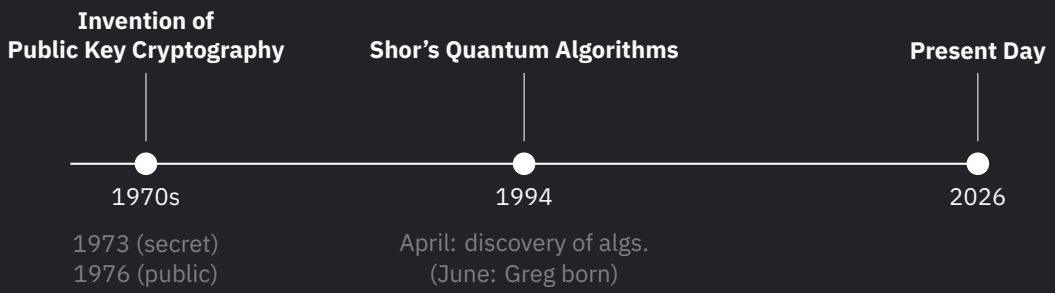


Timeline of public-key cryptography



Happy 32nd birthday to Shor's algorithms! 🥳

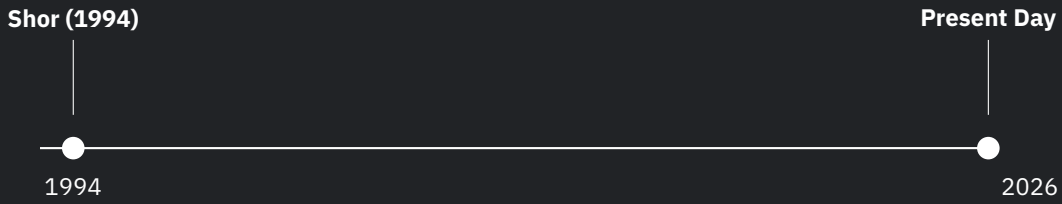
Timeline of public-key cryptography



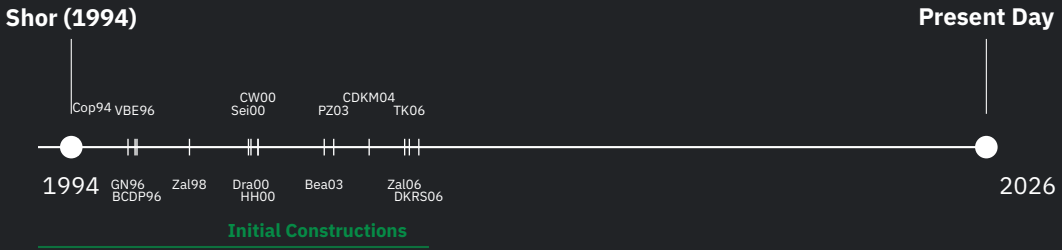
Happy 32nd birthday to Shor's algorithms! 🥳

Shor's discovery was closer to the invention of public-key cryptography than it is to the present day

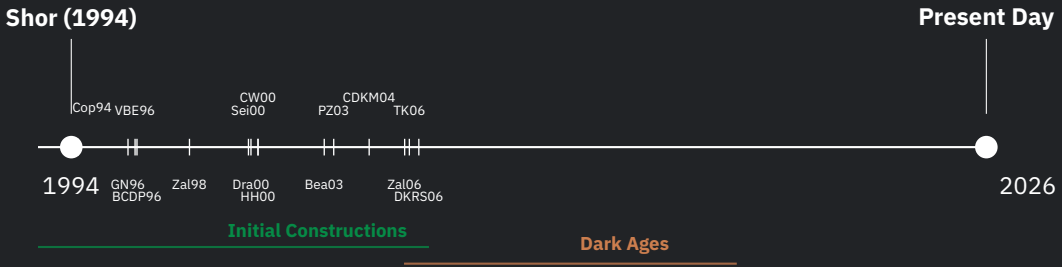
Timeline of quantum circuits for discrete log + factoring



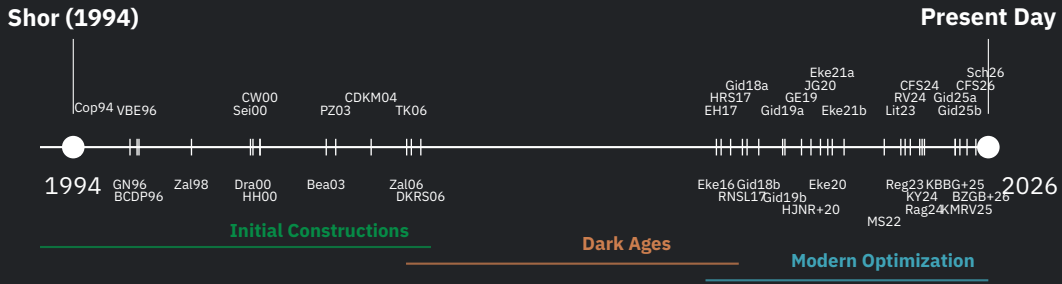
Timeline of quantum circuits for discrete log + factoring



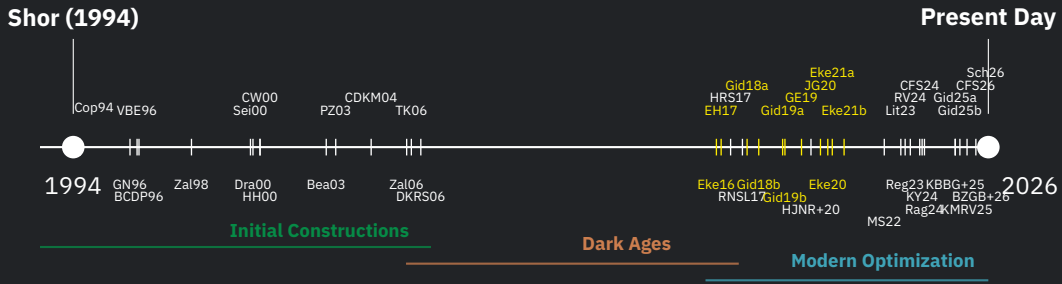
Timeline of quantum circuits for discrete log + factoring



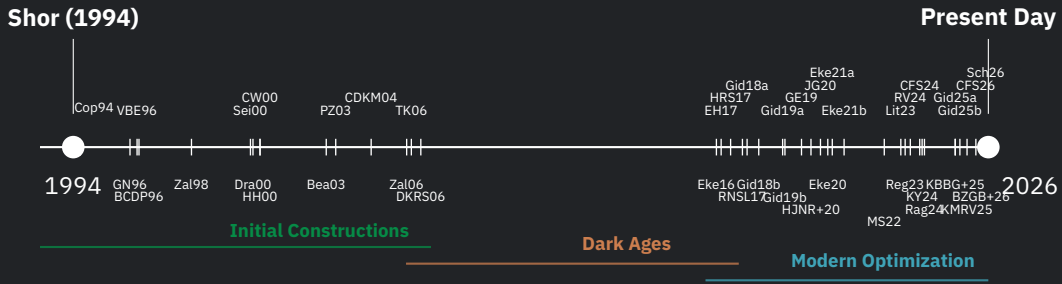
Timeline of quantum circuits for discrete log + factoring



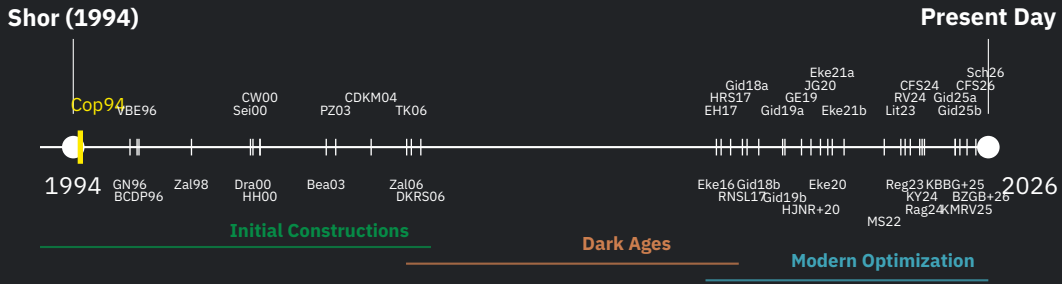
Timeline of quantum circuits for discrete log + factoring



Timeline of quantum circuits for discrete log + factoring

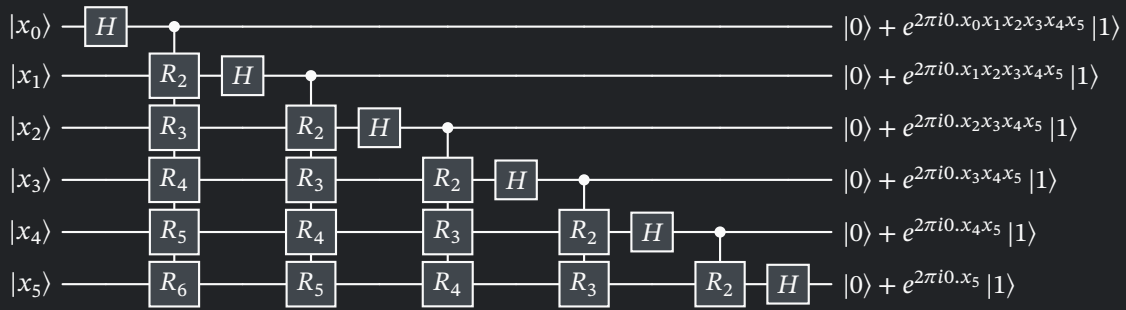


Timeline of quantum circuits for discrete log + factoring



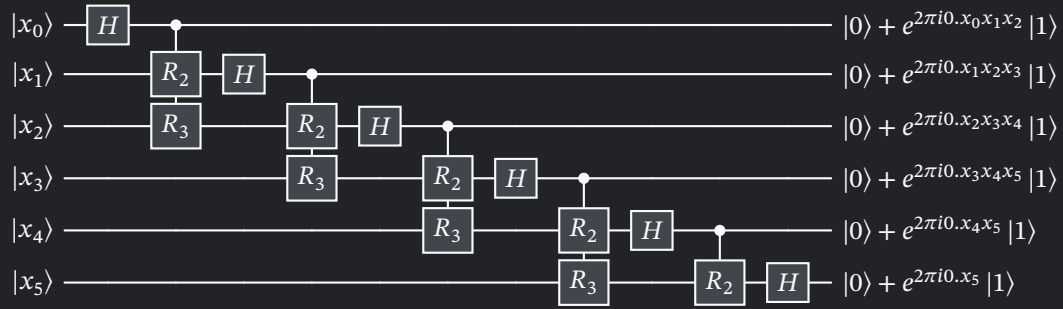
Coppersmith ’94: approximate quantum Fourier transform modulo 2^n

This is the “standard” QFT construction



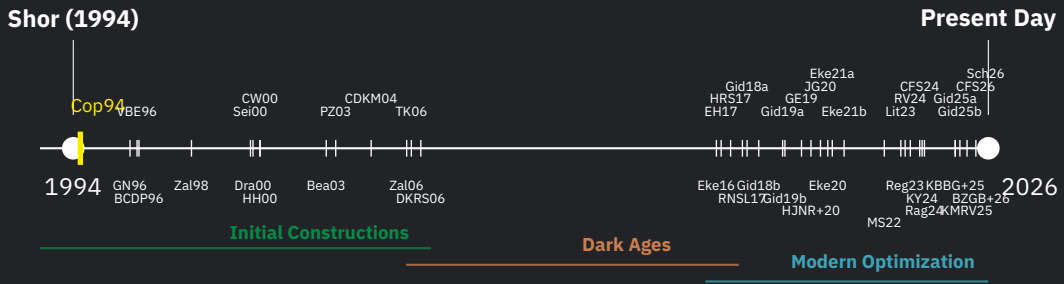
Coppersmith ’94: approximate QFT_{2^n}

Approximate QFT: drop small rotations!



Timeline of quantum circuits for discrete log + factoring

Timeline of quantum circuits for discrete log + factoring



Peter Shor didn't have the now-standard QFT_{2^n} construction for his original paper!

Peter Shor’s original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Observation:

- QFT_m for $m \lesssim O(\log n)$ is doable by "brute force" in $\mathrm{poly}(n)$ time

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Observation:

- QFT_m for $m \lesssim O(\log n)$ is doable by "brute force" in $\mathrm{poly}(n)$ time
- Maybe we can combine a bunch of tiny Fourier transforms into one big one?

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Observation:

- QFT_m for $m \lesssim O(\log n)$ is doable by "brute force" in $\mathrm{poly}(n)$ time
- Maybe we can combine a bunch of tiny Fourier transforms into one big one?
  - This is possible via a **Residue Number System (RNS)**.

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

Note: You will hear a lot more about RNS from André tomorrow at 11 AM!

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

To represent an n-bit integer x:

1. Pick a bunch of co-prime integers $\{m_i\}$, each of length $O(\log n)$

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

To represent an n-bit integer x:

1. Pick a bunch of co-prime integers $\{m_i\}$, each of length $O(\log n)$
   - Ensure their product $L = \prod_i m_i > x$

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

To represent an n-bit integer x:

1. Pick a bunch of co-prime integers $\{m_i\}$, each of length $O(\log n)$
   - Ensure their product $L = \prod_i m_i > x$
2. Store the tuple $(x_0, x_1, \cdots) = (x \bmod m_0, x \bmod m_1, \cdots)$

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

To represent an n-bit integer x:

1. Pick a bunch of co-prime integers $\{m_i\}$, each of length $O(\log n)$
   - Ensure their product $L = \prod_i m_i > x$
2. Store the tuple $(x_0, x_1, \cdots) = (x \bmod m_0, x \bmod m_1, \cdots)$

Aside: residue number systems (RNS)

A funky and neat alternative to binary representation!

To represent an n-bit integer x:

1. Pick a bunch of co-prime integers $\{m_i\}$, each of length $O(\log n)$
   - Ensure their product $L = \prod_i m_i > x$
2. Store the tuple $(x_0, x_1, \cdots) = (x \bmod m_0, x \bmod m_1, \cdots)$

Fact: *(Chinese Remainder Theorem)*

If $x < L$:

$$\sum_i x_i F_i \bmod L = x$$

(where F_i are some pre-computable integers that don't depend on x).

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$|x_{\text{RNS}}\rangle = \bigotimes_i |x_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$|x_{\text{RNS}}\rangle = \bigotimes_i |x_i\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i/L} |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\bigotimes_i \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i/L} |z_i\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i/L} |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z \left[\prod_i e^{2\pi i x_i z_i F_i / L} \right] \bigotimes_i |z_i\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i / L} |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z \left[\prod_i e^{2\pi i x_i z_i F_i / L} \right] |z_{\text{RNS}}\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i / L} |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z e^{2\pi i \sum_i x_i z_i F_i / L} |z_{\text{RNS}}\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i / L} |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z e^{2\pi i x z/L} \, |z_{\mathsf{RNS}}\rangle$$

Apply the following QFT-ish unitary to each:

$$|x_i\rangle \to \sum_{z_i=0}^{m_i-1} e^{2\pi i x_i z_i F_i/L} \, |z_i\rangle$$

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z e^{2\pi i x z / L} \, |z_{\text{RNS}}\rangle$$

Takeaway 1: We just did the QFT modulo L!

Peter Shor's original quantum Fourier transform (QFT)

Crucial question: can we efficiently perform the Fourier transform on the amplitudes of an n-qubit quantum state?

Keeping in mind $x_i = x \bmod m_i$

Consider the product state:

$$\sum_z e^{2\pi i x z / L} \, |z_{\text{RNS}}\rangle$$

Takeaway 2: Peter Shor is a really smart guy

SIMONS INSTITUTE
for the Theory of Computing

Thank you!

GDKM was supported by an appointment to the Intelligence Community Postdoctoral Research Fellowship Program at MIT administered by Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy and the Office of the Director of National Intelligence (ODNI).