GOOGLE QUANTUM AI



QCS GROUP MEETING

Lattice-based cryptography and (attempted) quantum attacks

GOOGLE QUANTUM AI
AUGUST 18, 2026

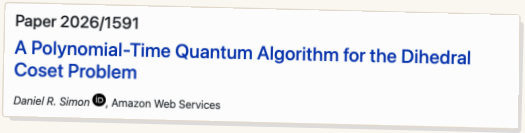
SEYOON RAGAVAN



> “Cryptographers seldom sleep well.”

SILVIO MICALI

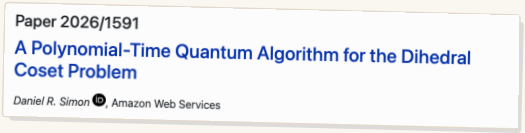
Two weeks ago...


Paper 2026/1591

A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem

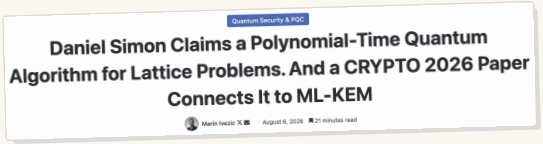
Daniel R. Simon, Amazon Web Services

Two weeks ago...


Paper 2026/1591

A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem

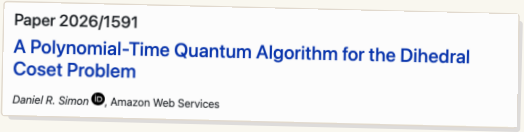
Daniel R. Simon, Amazon Web Services

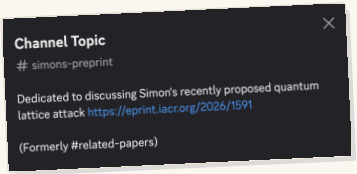

Quantum Security & PQC

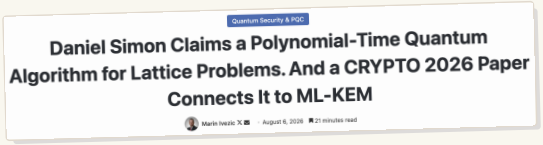
Daniel Simon Claims a Polynomial-Time Quantum Algorithm for Lattice Problems. And a CRYPTO 2026 Paper Connects It to ML-KEM

Marin Ivezic · August 6, 2026 · 21 minutes read

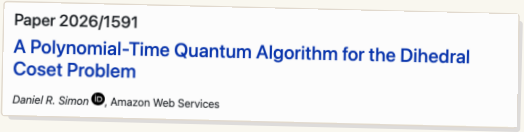
Two weeks ago...

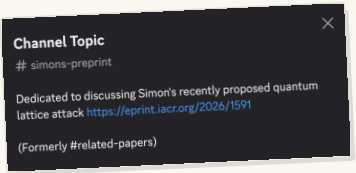


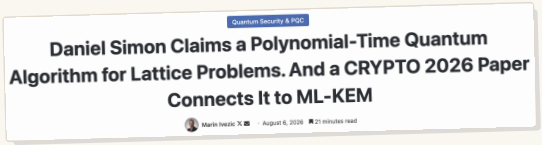


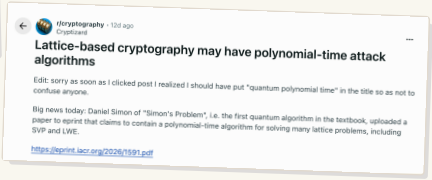


Two weeks ago...









… but PQC lives another day

Paper 2026/1693

The ePrint:2026/1591 Quantum Algorithm Does Not Solve DCP

Aparna Gupte, Massachusetts Institute of Technology
Seyoon Ragavan, Google (United States), Massachusetts Institute of Technology
Mark Zhandry, Google (United States), Stanford University

Lessons

- We should be wary of believing in the post-quantum security of current candidates

Lessons

- We should be wary of believing in the post-quantum security of current candidates
  - Post-quantum attack analysis is confined to pen and paper

Lessons

- We should be wary of believing in the post-quantum security of current candidates
  - Post-quantum attack analysis is confined to pen and paper
  - Even surprising classical attacks can arise!

ELEVEN DAYS BEFORE THIS TALK

Quasipolynomial Cryptanalysis of the McEliece Cryptosystem (or: PIR Meets McEliece)

Ashrujit Ghoshal* Yuval Ishai† Aayush Jain‡ Nuozhou Sun§

August 7, 2026

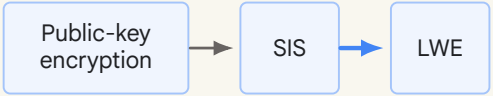
> "...a simple, classical quasipolynomial-time distinguisher for **Goppa–McEliece**"



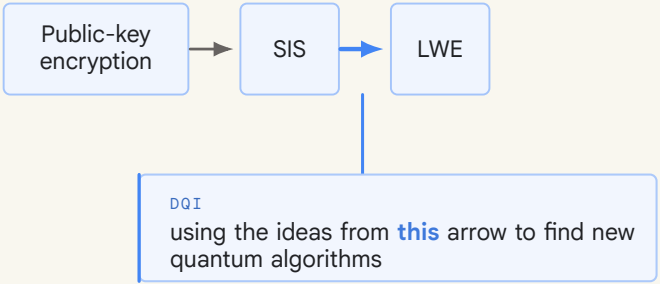
Lessons

- We should be wary of believing in the post-quantum security of current candidates
  - Post-quantum attack analysis is confined to pen and paper
  - Even surprising classical attacks can arise!
- No time like the present to study (quantum attacks on) lattice-based cryptography

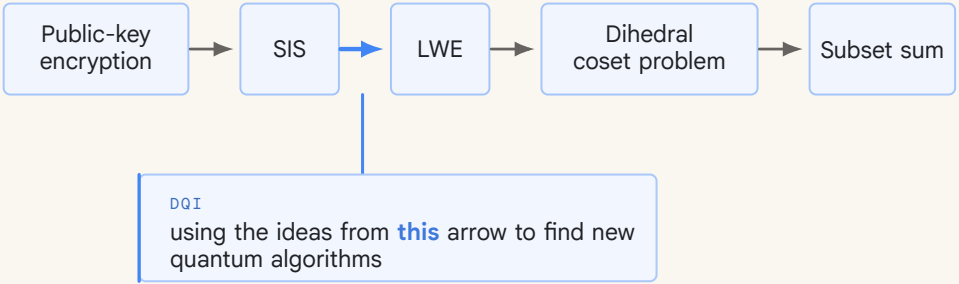
Roadmap



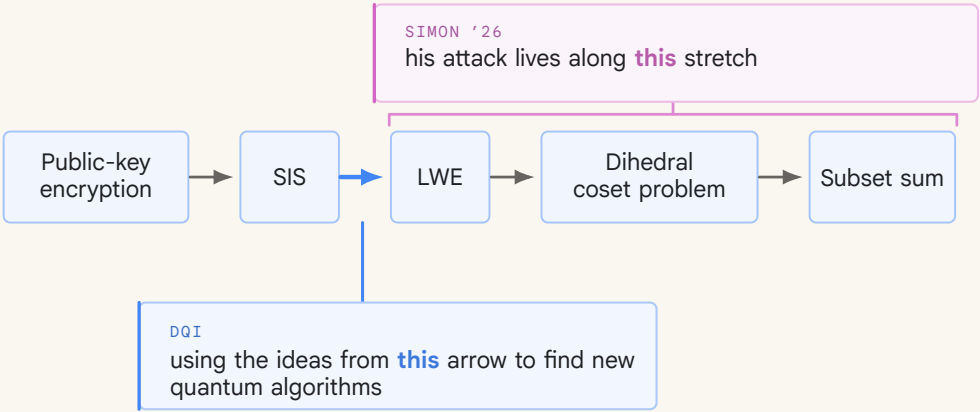
Roadmap


Public-key encryption
SIS
LWE
DQI
using the ideas from this arrow to find new quantum algorithms

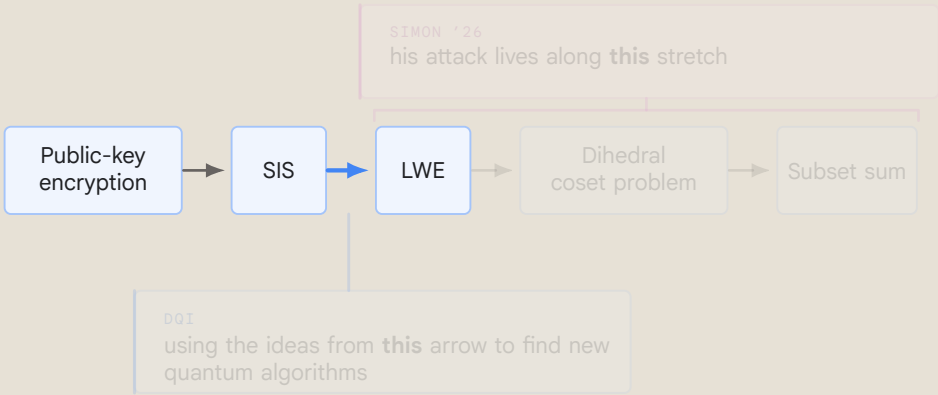
Roadmap


Public-key encryption
SIS
LWE
Dihedral coset problem
Subset sum
DQI
using the ideas from this arrow to find new quantum algorithms

Roadmap

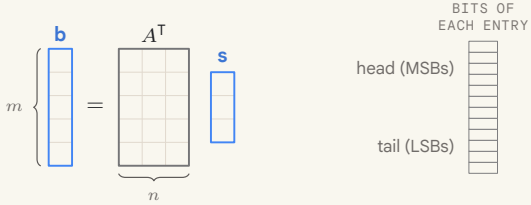

SIMON '26
his attack lives along this stretch
Public-key encryption
SIS
LWE
Dihedral coset problem
Subset sum
DQI
using the ideas from this arrow to find new quantum algorithms

Part 1: Lattice-based public-key cryptography



Linear systems of equations

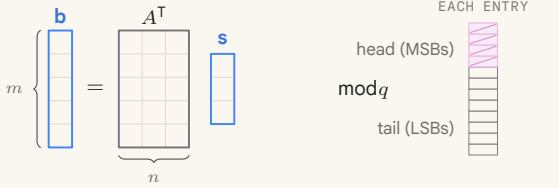
Baby problem: exact, over $\mathbb{R}$



read **s** straight off — **Gaussian elimination**

Linear systems of equations

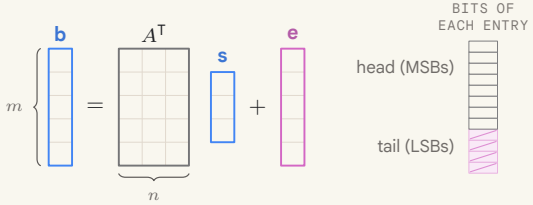
Attempt 1 — chop off the head: everything mod q



still **Gaussian elimination**, now over $\mathbb{Z}_q$

Linear systems of equations

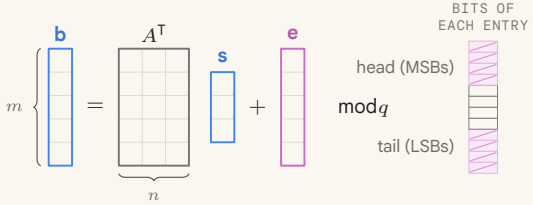
Attempt 2 — chop off the tail: add noise, over $\mathbb{R}$



least squares linear regression

Linear systems of equations

Chop off both:



nobody knows how (even with a quantum computer)

Only one corner is hard

EXACT, OVER $\mathbb{R}$

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s}$$

read **s** straight off:
Gaussian elimination



MOD q: CHOP THE HEAD

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} \bmod q$$

still **Gaussian elimination**, now over $\mathbb{Z}_q$



NOISE: CHOP THE TAIL

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e}$$

least squares — ordinary linear regression



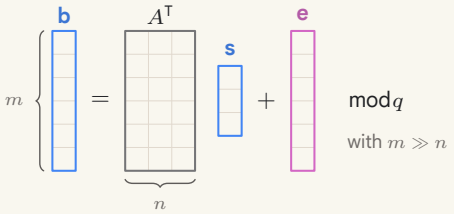
CHOP OFF BOTH

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e} \bmod q$$

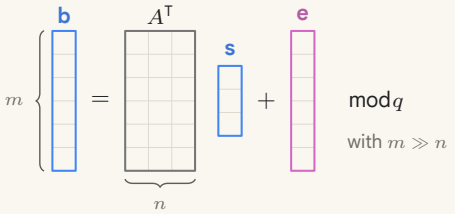
LWE, conjectured to be hard



Learning with errors (LWE) recap: decoding a random linear code from ℓ_2 errors

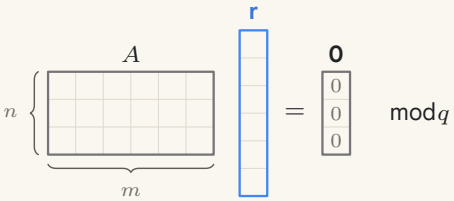


Learning with errors (LWE) recap: decoding a random linear code from ℓ_2 errors

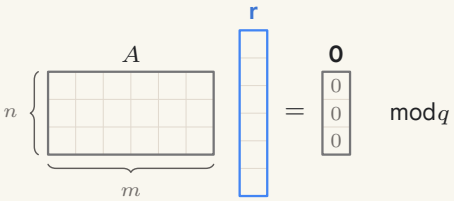


given A, recover **s** — where **e** is small in ℓ_2 norm, but unknown

Short integer solutions (SIS): finding a short parity check

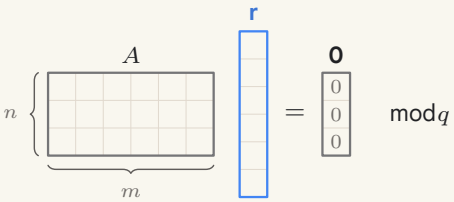


Short integer solutions (SIS): finding a short parity check



given A, find a **short** (low-norm) nonzero **r** that the code kills

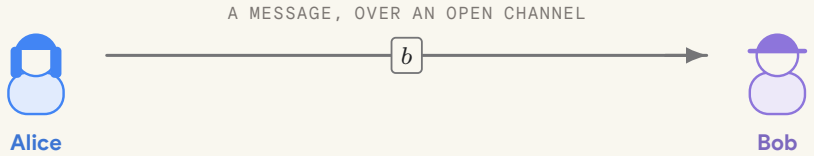
Short integer solutions (SIS): finding a short parity check



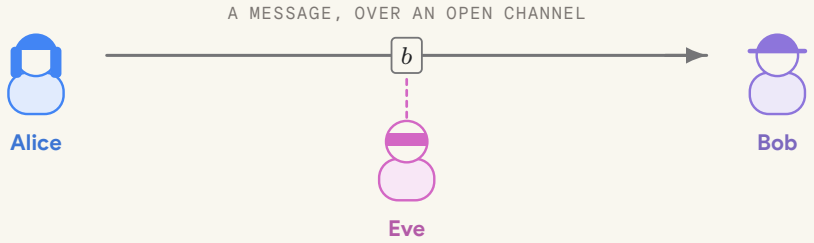
given A, find a **short** (low-norm) nonzero **r** that the code kills

> **Relevance is analogous to why low-weight parity checks are valuable for decoding — they limit noise propagation**

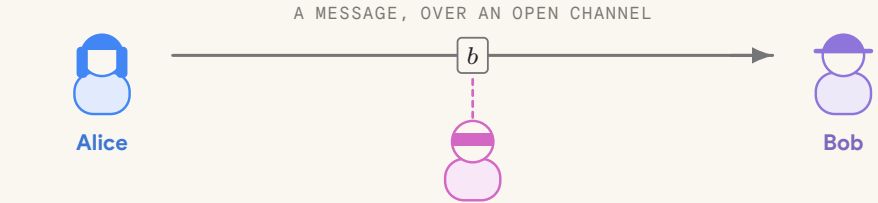
Alice sends Bob a bit b using LWE/SIS



Alice sends Bob a bit b using LWE/SIS



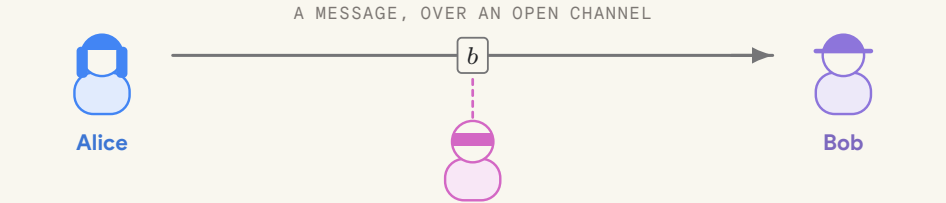
Alice sends Bob a bit b using LWE/SIS



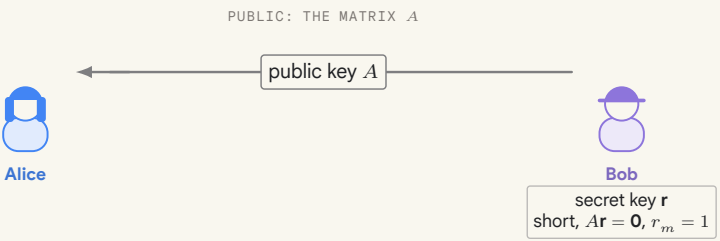
CHALLENGE 1

they have never met — there is no shared key

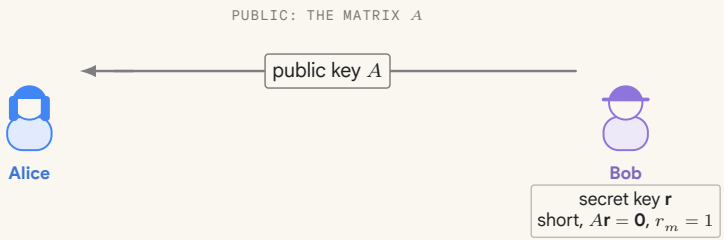
Alice sends Bob a bit b using LWE/SIS



Key generation

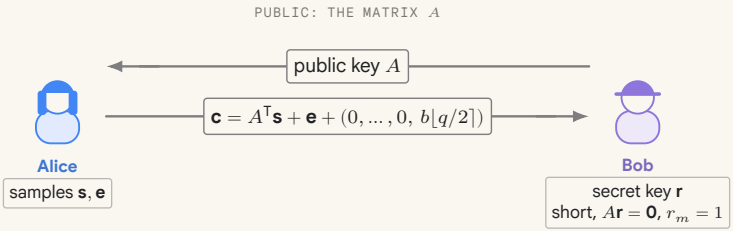


Key generation

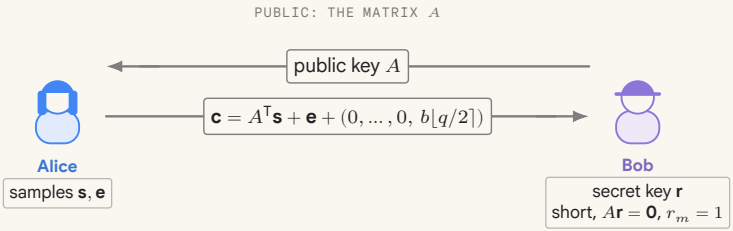


Public key A, secret key r — so solving SIS is equivalent to key recovery.

Encryption


PUBLIC: THE MATRIX A
public key A
$\mathbf{c} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e} + (0, \ldots, 0, b\lfloor q/2 \rceil)$
Alice
samples $\mathbf{s}, \mathbf{e}$
Bob
secret key $\mathbf{r}$
short, $A\mathbf{r} = \mathbf{0}$, $r_m = 1$

Encryption



The ciphertext is masked by $A^{\mathsf{T}}\mathbf{s} + \mathbf{e}$ — so solving LWE is equivalent to decryption.

Bob decrypts using the parity check **r**

$$\langle \mathbf{r}, \mathbf{c} \rangle \;=\; \langle \mathbf{r}, A^{\mathsf{T}}\mathbf{s} \rangle \;+\; \langle \mathbf{r}, \mathbf{e} \rangle \;+\; \langle \mathbf{r},\, b\lfloor q/2 \rceil (0, \ldots, 0, 1) \rangle$$

Bob decrypts using the parity check **r**

$$\langle \mathbf{r}, \mathbf{c} \rangle = \underbrace{\langle \mathbf{r}, A^{\mathsf{T}}\mathbf{s} \rangle}_{= 0 \text{ since } A\mathbf{r}=\mathbf{0}} + \langle \mathbf{r}, \mathbf{e} \rangle + \langle \mathbf{r}, b\lfloor q/2 \rceil (0, \dots, 0, 1) \rangle$$

Bob decrypts using the parity check **r**

$$\langle \mathbf{r}, \mathbf{c} \rangle = \underbrace{\langle \mathbf{r}, A^{\mathsf{T}}\mathbf{s} \rangle}_{= 0 \text{ since } A\mathbf{r}=\mathbf{0}} + \underbrace{\langle \mathbf{r}, \mathbf{e} \rangle}_{\text{small: both are short}} + \langle \mathbf{r},\, b\lfloor q/2 \rceil (0, \ldots, 0, 1) \rangle$$

Bob decrypts using the parity check **r**

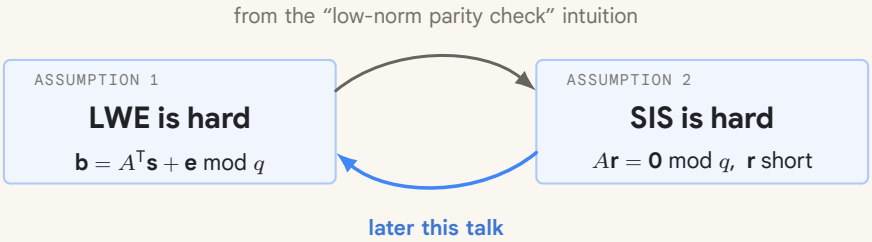
$$\langle \mathbf{r}, \mathbf{c} \rangle = \underbrace{\langle \mathbf{r}, A^{\mathsf{T}}\mathbf{s} \rangle}_{= 0 \text{ since } A\mathbf{r}=\mathbf{0}} + \underbrace{\langle \mathbf{r}, \mathbf{e} \rangle}_{\text{small: both are short}} + \underbrace{\langle \mathbf{r},\, b\lfloor q/2 \rceil (0, \ldots, 0, 1) \rangle}_{0 \text{ if } b=0,\ q/2 \text{ otherwise}}$$

Bob decrypts using the parity check **r**

$$\langle \mathbf{r}, \mathbf{c} \rangle = \underbrace{\langle \mathbf{r}, A^{\mathsf{T}}\mathbf{s} \rangle}_{=0 \text{ since } A\mathbf{r}=\mathbf{0}} + \underbrace{\langle \mathbf{r}, \mathbf{e} \rangle}_{\text{small: both are short}} + \underbrace{\langle \mathbf{r},\, b\lfloor q/2 \rceil (0, \ldots, 0, 1) \rangle}_{0 \text{ if } b=0,\ q/2 \text{ otherwise}}$$

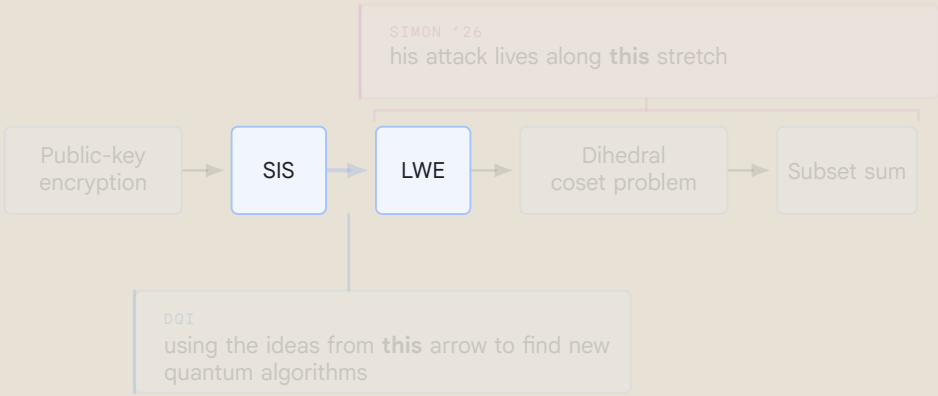
> **So $\langle \mathbf{r}, \mathbf{c} \rangle$ is $b\lfloor q/2 \rceil$ plus something small — round to the nearer of 0 and $q/2$ and read off b.**

The heart of lattice-based cryptography



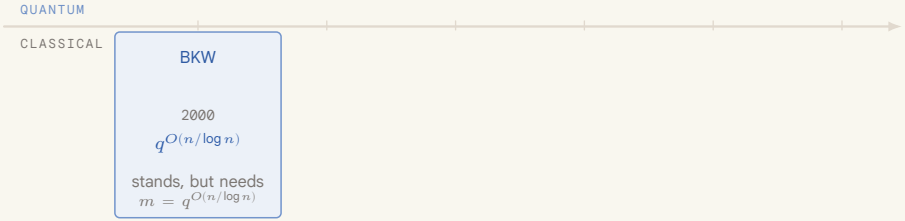
Attacking this encryption ⇔ solving one of these.

Part 2: An abbreviated history of LWE/SIS attacks

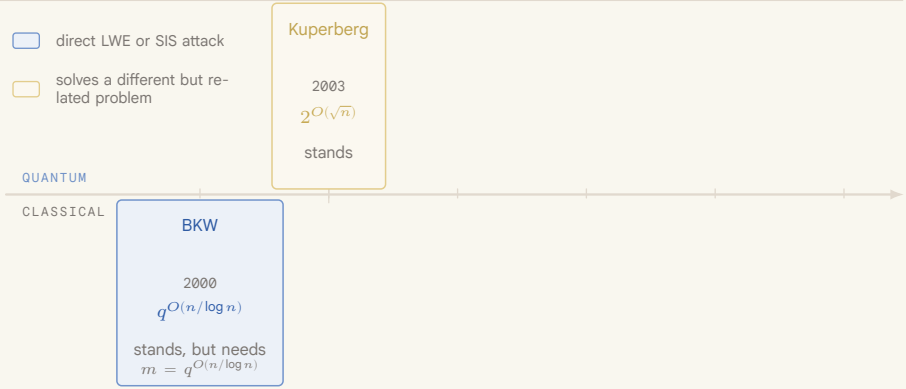


Timeline

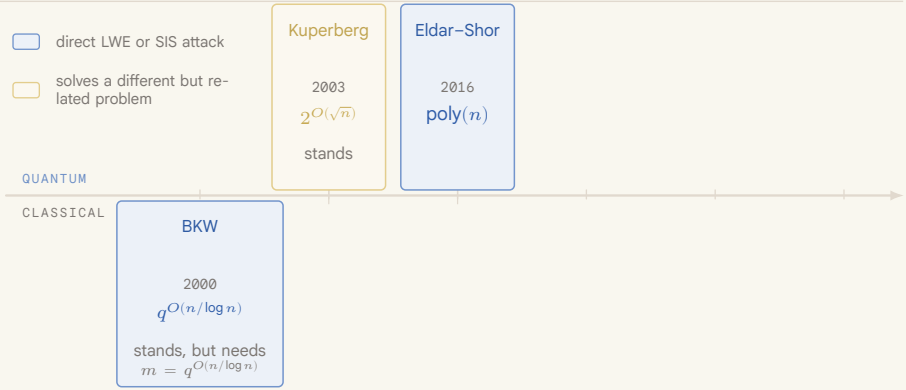
direct LWE or SIS attack



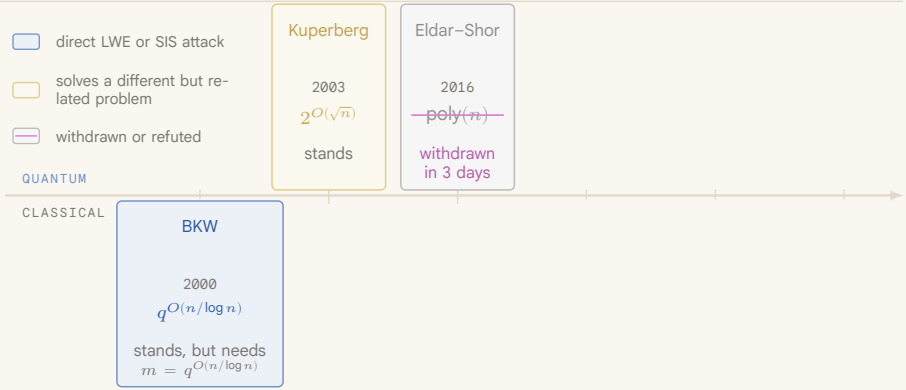
Timeline


direct LWE or SIS attack
solves a different but related problem
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
QUANTUM
CLASSICAL
BKW
2000
$q^{O(n/\log n)}$
stands, but needs
$m = q^{O(n/\log n)}$

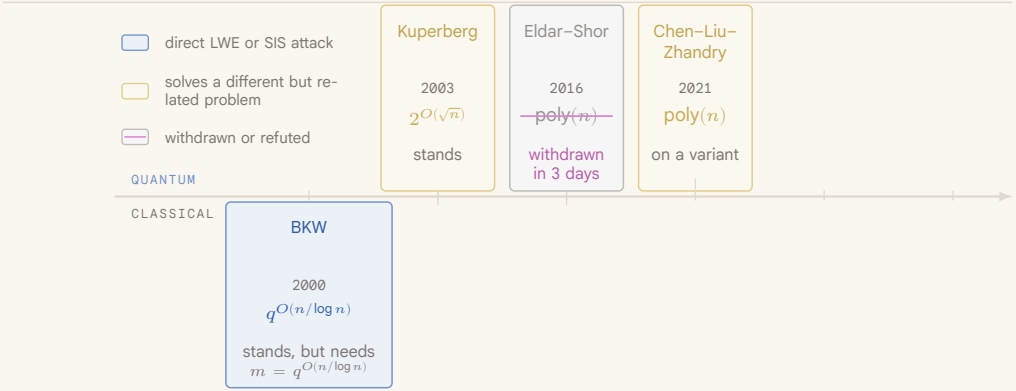
Timeline


direct LWE or SIS attack
solves a different but related problem
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
poly(n)
QUANTUM
CLASSICAL
BKW
2000
$q^{O(n/\log n)}$
stands, but needs
$m = q^{O(n/\log n)}$

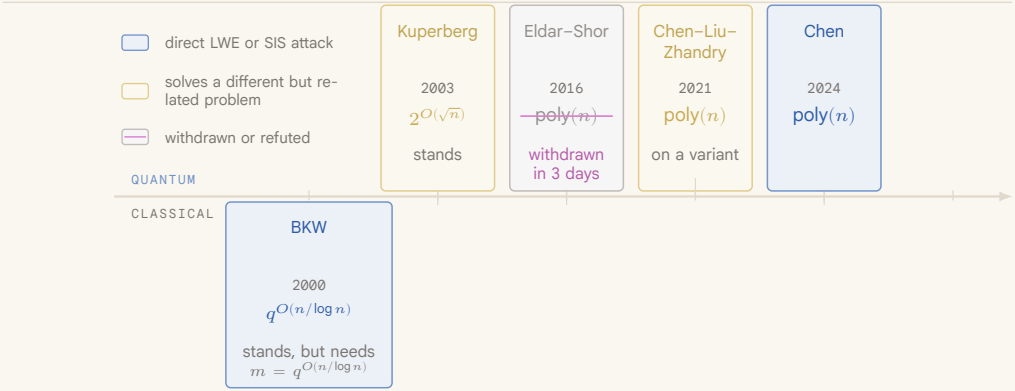
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
BKW
2000
$q^{O(n/\log n)}$
stands, but needs
$m = q^{O(n/\log n)}$

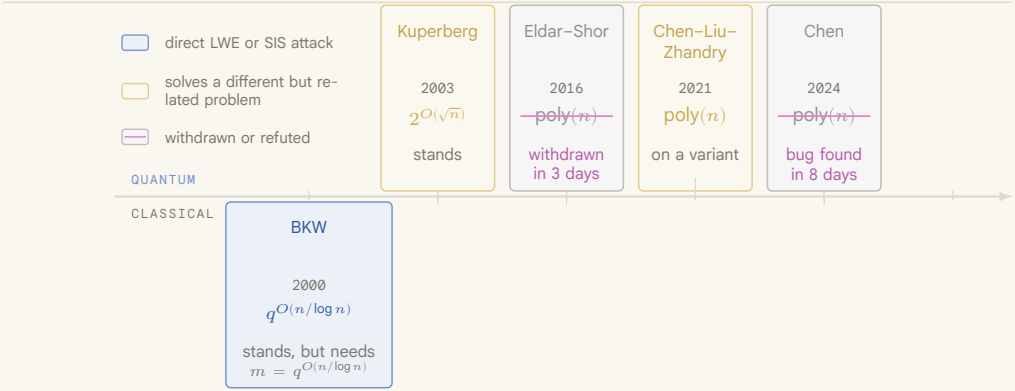
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
$\text{poly}(n)$
withdrawn in 3 days
Chen–Liu–Zhandry
2021
$\text{poly}(n)$
on a variant
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$

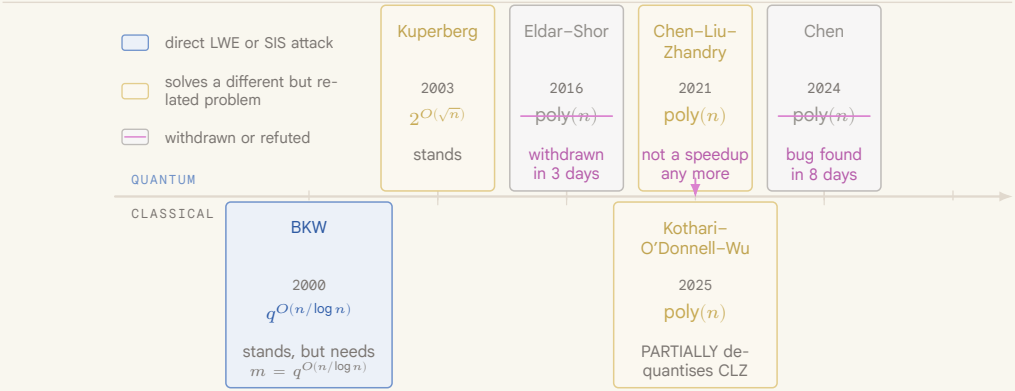
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
Chen–Liu–Zhandry
2021
poly(n)
on a variant
Chen
2024
poly(n)
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$

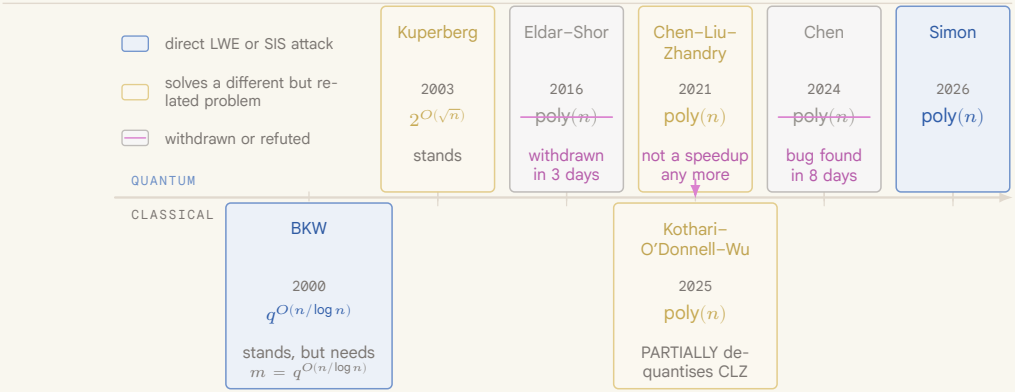
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
Chen–Liu–Zhandry
2021
poly(n)
on a variant
Chen
2024
~~poly(n)~~
bug found in 8 days
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$

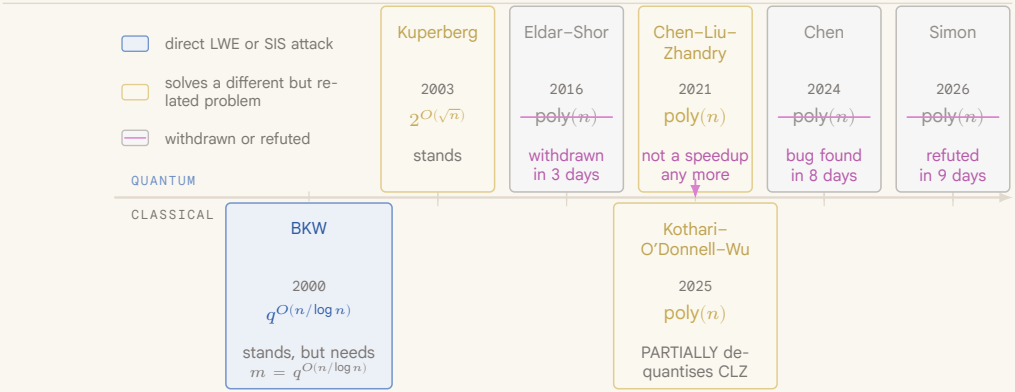
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
Chen–Liu–Zhandry
2021
poly(n)
not a speedup any more
Chen
2024
~~poly(n)~~
bug found in 8 days
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$
Kothari–O'Donnell–Wu
2025
poly(n)
PARTIALLY dequantises CLZ

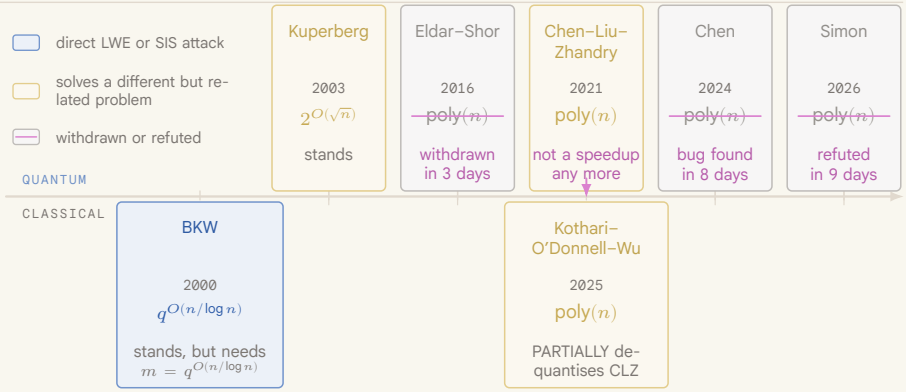
Timeline


direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
Chen–Liu–Zhandry
2021
poly(n)
not a speedup any more
Chen
2024
~~poly(n)~~
bug found in 8 days
Simon
2026
poly(n)
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$
Kothari–O'Donnell–Wu
2025
poly(n)
PARTIALLY dequantises CLZ

Timeline

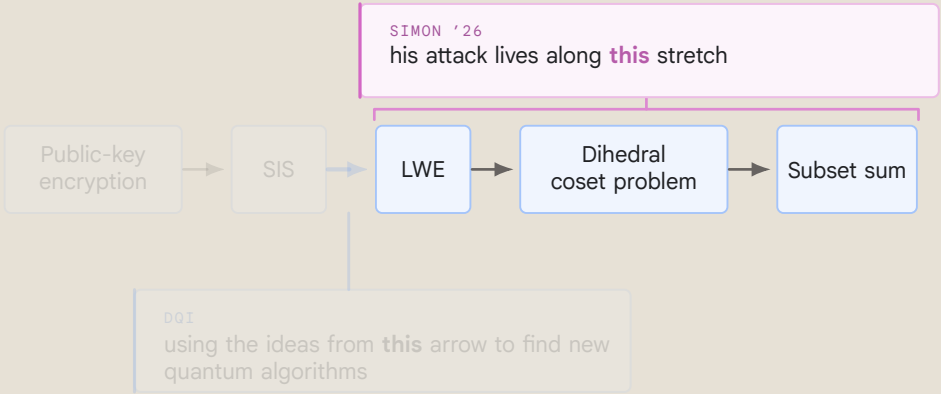

direct LWE or SIS attack
solves a different but related problem
withdrawn or refuted
QUANTUM
CLASSICAL
Kuperberg
2003
$2^{O(\sqrt{n})}$
stands
Eldar–Shor
2016
~~poly(n)~~
withdrawn in 3 days
Chen–Liu–Zhandry
2021
poly(n)
not a speedup any more
Chen
2024
~~poly(n)~~
bug found in 8 days
Simon
2026
~~poly(n)~~
refuted in 9 days
BKW
2000
$q^{O(n/\log n)}$
stands, but needs $m = q^{O(n/\log n)}$
Kothari–O'Donnell–Wu
2025
poly(n)
PARTIALLY dequantises CLZ

Timeline



The best we can do (classically or quantumly) is still $2^{O(n)}$.

Part 3: A dihedral vignette



Up next

Paper 2026/1591

A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem

Daniel R. Simon, Amazon Web Services

6 AUGUST 2026

Up next

Paper 2026/1591

A Polynomial-Time Quantum Algorithm for the Dihedral Coset Problem

Daniel R. Simon, Amazon Web Services

6 AUGUST 2026

Paper 2026/1693

The ePrint:2026/1591 Quantum Algorithm Does Not Solve DCP

Aparna Gupte, Massachusetts Institute of Technology
Seyoon Ragavan, Google (United States), Massachusetts Institute of Technology
Mark Zhandry, Google (United States), Stanford University

15 AUGUST 2026 · NINE DAYS LATER

The dihedral coset problem

Learn a hidden secret $s \in \mathbb{Z}_N$, given samples of the form

$$\frac{1}{\sqrt{2}}\big(|x\rangle|0\rangle + |x+s\rangle|1\rangle\big)$$

for uniform but unknown x — a fresh one every sample

The dihedral coset problem

Learn a hidden secret $s \in \mathbb{Z}_N$, given samples of the form

$$\frac{1}{\sqrt{2}}\left(|x\rangle|0\rangle + |x+s\rangle|1\rangle\right)$$

for uniform but unknown x — a fresh one every sample



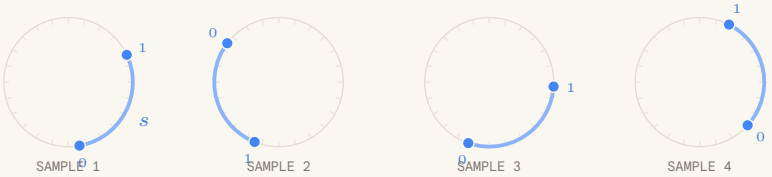
the pair moves every time — the gap never does

The dihedral coset problem

Learn a hidden secret $s \in \mathbb{Z}_N$, given samples of the form

$$\frac{1}{\sqrt{2}}\left(|x\rangle|0\rangle + |x+s\rangle|1\rangle\right)$$

for uniform but unknown x — a fresh one every sample



the pair moves every time — the gap never does

Regev's 2004 reduction: breaking LWE reduces to solving DCP.

Fourier sampling

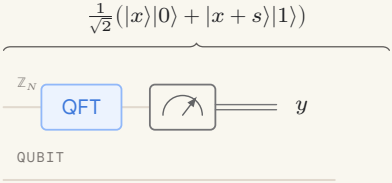
$$\frac{1}{\sqrt{2}}(|x\rangle|0\rangle + |x+s\rangle|1\rangle)$$

$\mathbb{Z}_N$

QUBIT

Fourier sampling

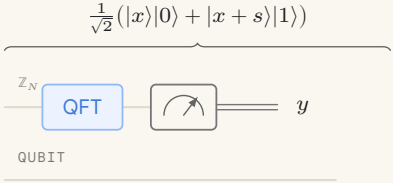
$$\mathrm{QFT}\,|x\rangle = \tfrac{1}{\sqrt{N}} \sum_y \omega_N^{xy}\,|y\rangle$$



Fourier sampling

$$\text{QFT}\,|x\rangle = \tfrac{1}{\sqrt{N}} \sum_y \omega_N^{xy}\,|y\rangle$$

so the state is $\tfrac{1}{\sqrt{2N}} \sum_y \omega_N^{xy}\,|y\rangle(|0\rangle + \omega_N^{ys}|1\rangle)$



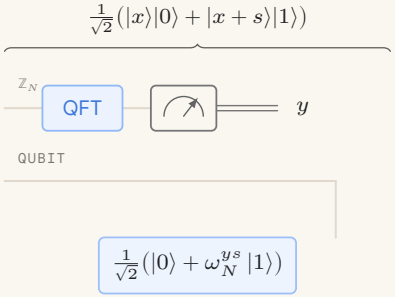
Fourier sampling

$$\text{QFT}\,|x\rangle = \tfrac{1}{\sqrt{N}} \sum_y \omega_N^{xy}\,|y\rangle$$

so the state is $\tfrac{1}{\sqrt{2N}} \sum_y \omega_N^{xy}\,|y\rangle(|0\rangle + \omega_N^{ys}|1\rangle)$

$$\xrightarrow{\text{measure } y} \tfrac{1}{\sqrt{2}}(|0\rangle + \omega_N^{ys}|1\rangle)$$

up to the global phase ω_N^{xy}



Regev’s 2004 reduction to subset sum

FROM k FOURIER SAMPLES, WITH $\mathbf{y}$ KNOWN

$$\bigotimes_{i=1}^{k}\left(|0\rangle+\omega_N^{y_i s}|1\rangle\right) \;=\; \sum_{\mathbf{b}\in\{0,1\}^k} \omega_N^{s\langle \mathbf{b},\mathbf{y}\rangle}\,|\mathbf{b}\rangle$$

Regev's 2004 reduction to subset sum

FROM k FOURIER SAMPLES, WITH $\mathbf{y}$ KNOWN

$$\bigotimes_{i=1}^{k} (|0\rangle + \omega_N^{y_i s} |1\rangle) \;=\; \sum_{\mathbf{b} \in \{0,1\}^k} \omega_N^{s\langle \mathbf{b}, \mathbf{y} \rangle} \, |\mathbf{b}\rangle$$

COMPUTE $\langle \mathbf{b}, \mathbf{y} \rangle$ COHERENTLY

$$\sum_{\mathbf{b}} \omega_N^{s\langle \mathbf{b}, \mathbf{y} \rangle} \, |\mathbf{b}\rangle \, |\langle \mathbf{b}, \mathbf{y} \rangle\rangle$$

Regev's 2004 reduction to subset sum

FROM k FOURIER SAMPLES, WITH $\mathbf{y}$ KNOWN

$$\bigotimes_{i=1}^{k}(|0\rangle + \omega_N^{y_i s}|1\rangle) \;=\; \sum_{\mathbf{b}\in\{0,1\}^k} \omega_N^{s\langle \mathbf{b},\mathbf{y}\rangle}\,|\mathbf{b}\rangle$$

COMPUTE $\langle \mathbf{b},\mathbf{y}\rangle$ COHERENTLY

$$\sum_{\mathbf{b}} \omega_N^{s\langle \mathbf{b},\mathbf{y}\rangle}\,|\mathbf{b}\rangle\,|\langle \mathbf{b},\mathbf{y}\rangle\rangle$$

IF WE COULD UNCOMPUTE $\mathbf{b}$…

$$\sum_z \omega_N^{sz}\,|z\rangle \;=\; \mathsf{QFT}\,|s\rangle$$

…and we would be done

Regev's 2004 reduction to subset sum

FROM k FOURIER SAMPLES, WITH $\mathbf{y}$ KNOWN

$$\bigotimes_{i=1}^{k}(|0\rangle + \omega_N^{y_i s}|1\rangle) \;=\; \sum_{\mathbf{b}\in\{0,1\}^k} \omega_N^{s\langle \mathbf{b},\mathbf{y}\rangle}\,|\mathbf{b}\rangle$$

↓ COMPUTE $\langle \mathbf{b},\mathbf{y}\rangle$ COHERENTLY

$$\sum_{\mathbf{b}} \omega_N^{s\langle \mathbf{b},\mathbf{y}\rangle}\,|\mathbf{b}\rangle\,|\langle \mathbf{b},\mathbf{y}\rangle\rangle$$

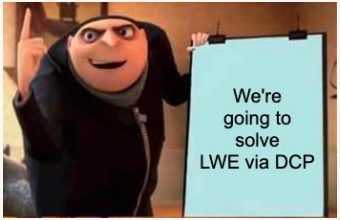
↓ IF WE COULD UNCOMPUTE $\mathbf{b}$…

$$\sum_z \omega_N^{sz}\,|z\rangle \;=\; \mathrm{QFT}\,|s\rangle$$

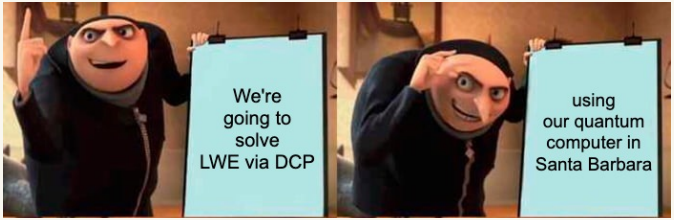
…and we would be done

but undoing $\mathbf{b} \mapsto \langle \mathbf{b},\mathbf{y}\rangle$ is **exactly subset sum**
— which is believed to be hard

Regev ’04: recap



Regev ’04: recap



Regev '04: recap



Regev ’04: recap



Regev ’04: recap



Simon’s idea: keep the Regev ’04 template, and aim to “solve subset sum erroneously”.

Simon’s two ideas

IDEA 1 — ALSO IN REGEV ’04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

Simon's two ideas

IDEA 1 — ALSO IN REGEV '04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

$$\omega_N^{st} = \omega_N^{s\, t_{\text{low}}} \cdot (-1)^{s \cdot \text{MSB}(t)} \qquad \Longrightarrow \qquad \sum_{\mathbf{b}} (-1)^{s \cdot \text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)} |\mathbf{b}\rangle$$

Only the **top bit** of the subset sum still matters — and only the **lowest bit of** s.

Simon's two ideas

IDEA 1 — ALSO IN REGEV '04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

$$\omega_N^{st} = \omega_N^{s\, t_{\text{low}}} \cdot (-1)^{s \cdot \text{MSB}(t)} \qquad \Longrightarrow \qquad \sum_{\mathbf{b}} (-1)^{s \cdot \text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)} |\mathbf{b}\rangle$$

Only the **top bit** of the subset sum still matters — and only the **lowest bit of** s.

IDEA 2

The **b** register is still there, so we still have to produce $\text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)$ from **b** — but that bit survives a lot of forgetting.

Group the coordinates into small buckets and keep only enough high-order bits of each bucket's own subset sum.



Simon's two ideas

IDEA 1 — ALSO IN REGEV '04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

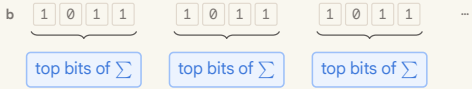
$$\omega_N^{st} = \omega_N^{s\, t_{\text{low}}} \cdot (-1)^{s \cdot \text{MSB}(t)} \qquad \Longrightarrow \qquad \sum_{\mathbf{b}} (-1)^{s \cdot \text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)} |\mathbf{b}\rangle$$

Only the **top bit** of the subset sum still matters — and only the **lowest bit of** s.

IDEA 2

The **b** register is still there, so we still have to produce $\text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)$ from **b** — but that bit survives a lot of forgetting.

Group the coordinates into small buckets and keep only enough high-order bits of each bucket's own subset sum.



Simon's two ideas

IDEA 1 — ALSO IN REGEV '04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

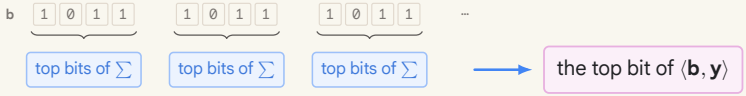
$$\omega_N^{st} = \omega_N^{s\, t_{\text{low}}} \cdot (-1)^{s \cdot \text{MSB}(t)} \qquad \Longrightarrow \qquad \sum_{\mathbf{b}} (-1)^{s \cdot \text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)} |\mathbf{b}\rangle$$

Only the **top bit** of the subset sum still matters — and only the **lowest bit of** s.

IDEA 2

The **b** register is still there, so we still have to produce $\text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)$ from **b** — but that bit survives a lot of forgetting.

Group the coordinates into small buckets and keep only enough high-order bits of each bucket's own subset sum.



Simon's two ideas

IDEA 1 — ALSO IN REGEV '04, FOR TECHNICAL REASONS

Measure all but the highest-order bit of $t = \langle \mathbf{b}, \mathbf{y} \rangle$. Splitting $t = t_{\text{low}} + \text{MSB}(t) \cdot N/2$,

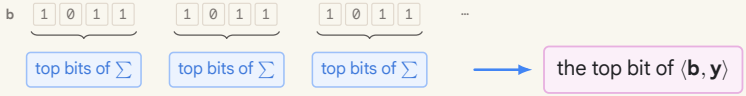
$$\omega_N^{st} = \omega_N^{s\, t_{\text{low}}} \cdot (-1)^{s \cdot \text{MSB}(t)} \qquad \Longrightarrow \qquad \sum_{\mathbf{b}} (-1)^{s \cdot \text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)} |\mathbf{b}\rangle$$

Only the **top bit** of the subset sum still matters — and only the **lowest bit of** s.

IDEA 2

The **b** register is still there, so we still have to produce $\text{MSB}(\langle \mathbf{b}, \mathbf{y} \rangle)$ from **b** — but that bit survives a **lot of forgetting**.

Group the coordinates into small buckets and keep only enough high-order bits of each bucket's own subset sum.



Why this template cannot work

OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

Why this template cannot work

OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

AND THAT IS FATAL

Doing so **effectively** collapses the $|\mathbf{b}\rangle$ register to a **singleton** — and a singleton is useless: there is no superposition left to interfere.

Why this template cannot work

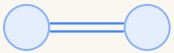
OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

AND THAT IS FATAL

Doing so **effectively** collapses the $|\mathbf{b}\rangle$ register to a **singleton** — and a singleton is useless: there is no superposition left to interfere.

THE INTUITION



an EPR pair

Why this template cannot work

OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

AND THAT IS FATAL

Doing so **effectively** collapses the $|\mathbf{b}\rangle$ register to a **singleton** — and a singleton is useless: there is no superposition left to interfere.

THE INTUITION



Why this template cannot work

OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

AND THAT IS FATAL

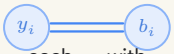
Doing so **effectively** collapses the $|\mathbf{b}\rangle$ register to a **singleton** — and a singleton is useless: there is no superposition left to interfere.

THE INTUITION



an EPR pair

what you kept is a **singleton**



each y_i with its own b_i

Why this template cannot work

OBSERVATION

That same per-bucket construction can be carried out while **forgetting most of y**.

AND THAT IS FATAL

Doing so **effectively** collapses the $|\mathbf{b}\rangle$ register to a **singleton** — and a singleton is useless: there is no superposition left to interfere.

THE INTUITION





knowing all but one bit of the overall sum
and that $b_i = 1$ pins down y_i



Takeaways

THE NO-GO

- Regev's '04 template forces your hand
- You must use **basically all** of **y** in the subset-sum stage
- The more you forget of it, the more $|\mathbf{b}\rangle$ collapses

Takeaways

THE NO-GO

- Regev's '04 template forces your hand
- You must use **basically all** of **y** in the subset-sum stage
- The more you forget of it, the more $|\mathbf{b}\rangle$ collapses

STILL WORTH HAVING

- Simon's small per-bucket caches
- Top bit of the subset sum, from very little
- May yet be useful — here or elsewhere

Takeaways

THE NO-GO

- Regev's '04 template forces your hand
- You must use **basically all** of **y** in the subset-sum stage
- The more you forget of it, the more $|\mathbf{b}\rangle$ collapses

STILL WORTH HAVING

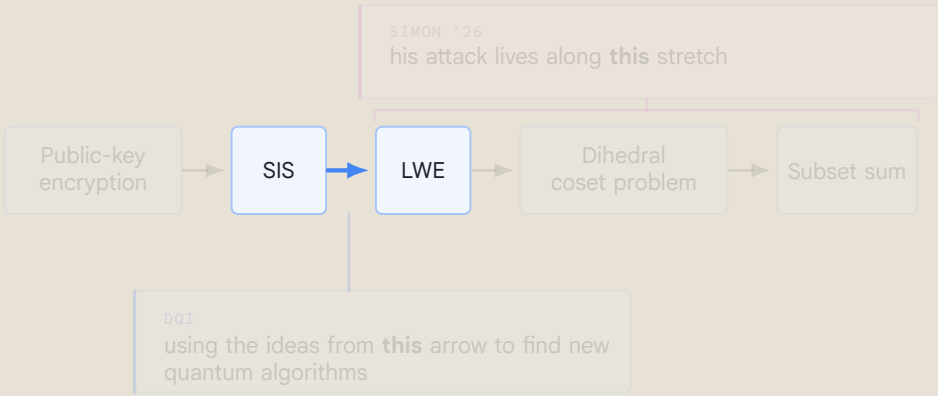
- Simon's small per-bucket caches
- Top bit of the subset sum, from very little
- May yet be useful — here or elsewhere

AND A WAY TO READ SIMON'S MANUSCRIPT

- Hunt the red herrings, strip them away
- Steps that provably cannot affect correctness
- e.g. many Hadamard-basis measurements — only **"all zeros or not"** is ever used

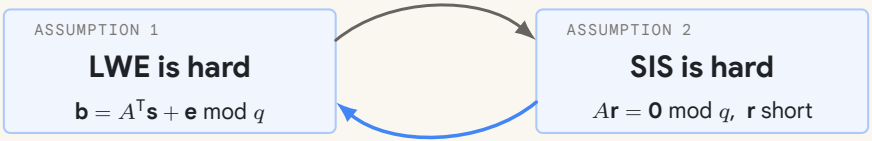
$\rightarrow$ cuts through a lot of noise

Part 4: Regev's 2005 reduction



The heart of lattice-based cryptography

from the "low-norm parity check" intuition



The Fourier convolution theorem

$f(\mathbf{x}) = \mathbb{1}[A\mathbf{x} = \mathbf{0}]$

PRIMAL

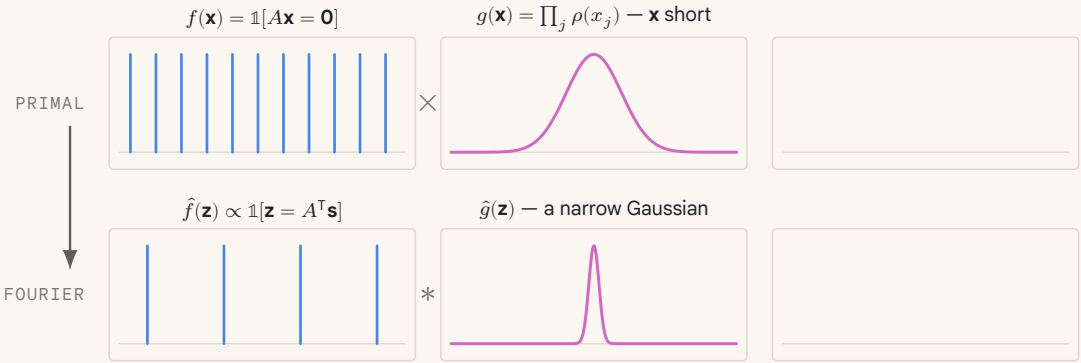


$\hat{f}(\mathbf{z}) \propto \mathbb{1}[\mathbf{z} = A^{\mathsf{T}}\mathbf{s}]$

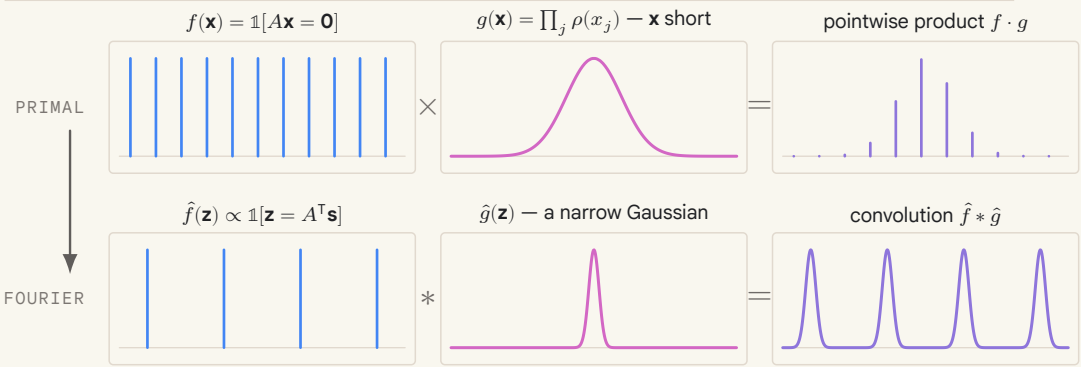
FOURIER



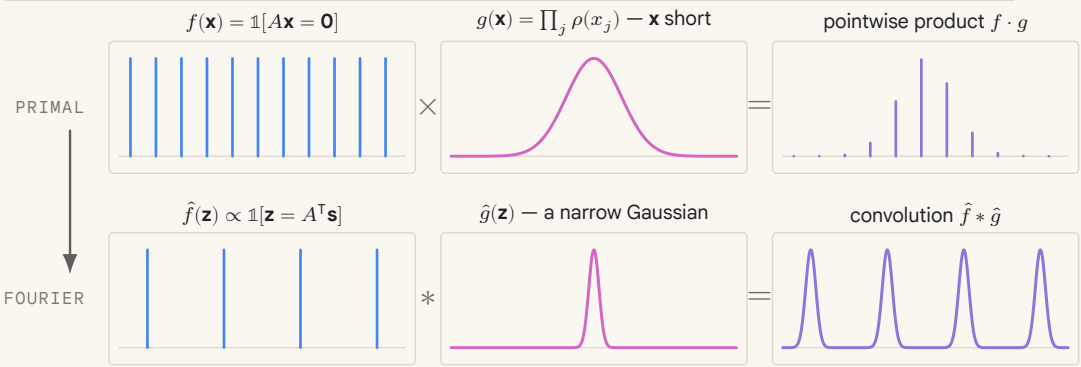
The Fourier convolution theorem



The Fourier convolution theorem



The Fourier convolution theorem



Multiplying two functions pointwise is convolving their Fourier transforms.

What we can build, and what we want

GOAL

Output an **x** with $A\mathbf{x} = \mathbf{0}$ **and x** short — a SIS solution.

$|\ker A\rangle = \sum_{\mathbf{x}} f(\mathbf{x})|\mathbf{x}\rangle$

☑ easy to prepare

$|\rho\rangle^{\otimes m} = \sum_{\mathbf{x}} g(\mathbf{x})|\mathbf{x}\rangle$

☑ easy to prepare: one Gaussian, m times

What we can build, and what we want

GOAL

Output an **x** with $A\mathbf{x} = \mathbf{0}$ **and x** short — a SIS solution.

$|\ker A\rangle = \sum_{\mathbf{x}} f(\mathbf{x})|\mathbf{x}\rangle$

☑ easo to prepare

$|\rho\rangle^{\otimes m} = \sum_{\mathbf{x}} g(\mathbf{x})|\mathbf{x}\rangle$

☑ easo to prepare: one Gaussian, m times

$\odot$

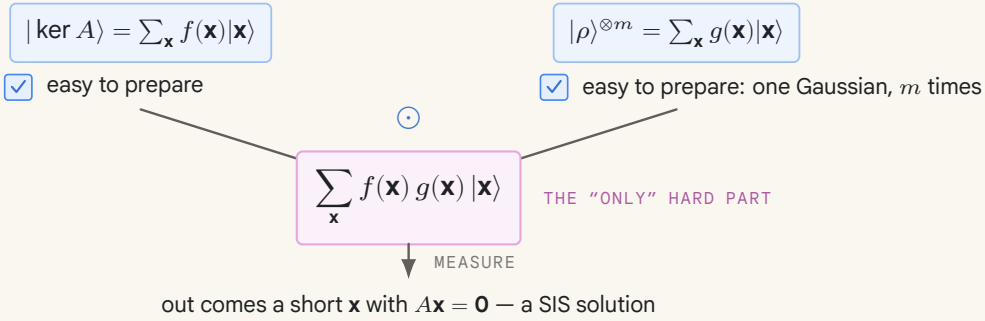
$$\sum_{\mathbf{x}} f(\mathbf{x})\, g(\mathbf{x})\, |\mathbf{x}\rangle$$

THE "ONLY" HARD PART

What we can build, and what we want

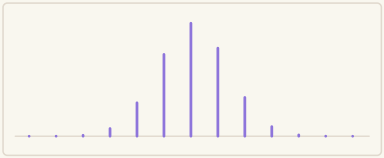
GOAL

Output an **x** with $A\mathbf{x} = \mathbf{0}$ **and x** short — a SIS solution.



Take the QFT

what we want



$\sum_{\mathbf{x}} f(\mathbf{x}) g(\mathbf{x}) |\mathbf{x}\rangle$

QFT

so it is enough to prepare



$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e}) \, |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$

Building the convolution

- Prepare $\sum_{\mathbf{s}} |A^{\mathsf{T}}\mathbf{s}\rangle$ and $\sum_{\mathbf{e}} \hat{g}(\mathbf{e})|\mathbf{e}\rangle$ side by side — two registers, two unrelated pictures



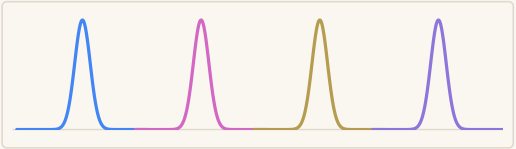
$\sum_{\mathbf{s}} |A^{\mathsf{T}}\mathbf{s}\rangle$



$\sum_{\mathbf{e}} \hat{g}(\mathbf{e})|\mathbf{e}\rangle$

Building the convolution

- Prepare $\sum_{\mathbf{s}} |A^{\mathsf{T}}\mathbf{s}\rangle$ and $\sum_{\mathbf{e}} \hat{g}(\mathbf{e})|\mathbf{e}\rangle$ side by side — two registers, two unrelated pictures
- Add them coherently $\rightarrow$ a **tagged** convolution



$$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e})\, |A^{\mathsf{T}}\mathbf{s}\rangle |\mathbf{e}\rangle |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$$

RIGHT SHAPE, BUT EVERY PIECE IS TAGGED

Building the convolution

- Prepare $\sum_{\mathbf{s}} |A^{\mathsf{T}}\mathbf{s}\rangle$ and $\sum_{\mathbf{e}} \hat{g}(\mathbf{e})|\mathbf{e}\rangle$ side by side — two registers, two unrelated pictures
- Add them coherently $\rightarrow$ a **tagged** convolution
- The decoding problem we are left with: **uncompute** the tag $A^{\mathsf{T}}\mathbf{s}$

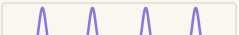


$$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e}) \, |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$$

TAG FORGOTTEN — ONE PICTURE

Voilà! The decoding problem is exactly LWE!

WHAT WE BUILT



$$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e}) \, |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$$

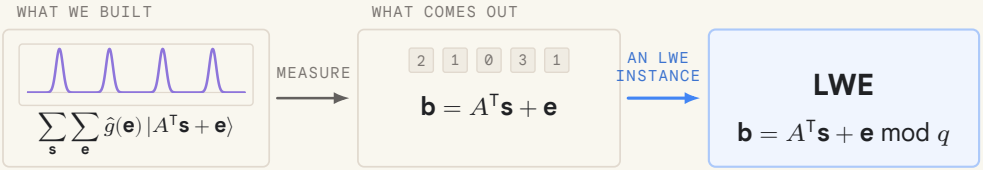
MEASURE

WHAT COMES OUT

2 1 0 3 1

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e}$$

Voilà! The decoding problem is exactly LWE!



Voilà! The decoding problem is exactly LWE!

WHAT WE BUILT



$$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e}) \, |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$$

MEASURE

WHAT COMES OUT

2 1 0 3 1

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e}$$

AN LWE INSTANCE

LWE

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e} \bmod q$$

WHAT THAT BUYS US

an LWE solver

USED AS THE DECODER

a short **x** with $A\mathbf{x} = \mathbf{0}$

Voilà! The decoding problem is exactly LWE!

WHAT WE BUILT



$$\sum_{\mathbf{s}} \sum_{\mathbf{e}} \hat{g}(\mathbf{e}) \, |A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$$

MEASURE

WHAT COMES OUT

2 1 0 3 1

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e}$$

AN LWE INSTANCE

LWE

$$\mathbf{b} = A^{\mathsf{T}}\mathbf{s} + \mathbf{e} \bmod q$$

WHAT THAT BUYS US

an LWE solver — USED AS THE DECODER → a short **x** with $A\mathbf{x} = \mathbf{0}$

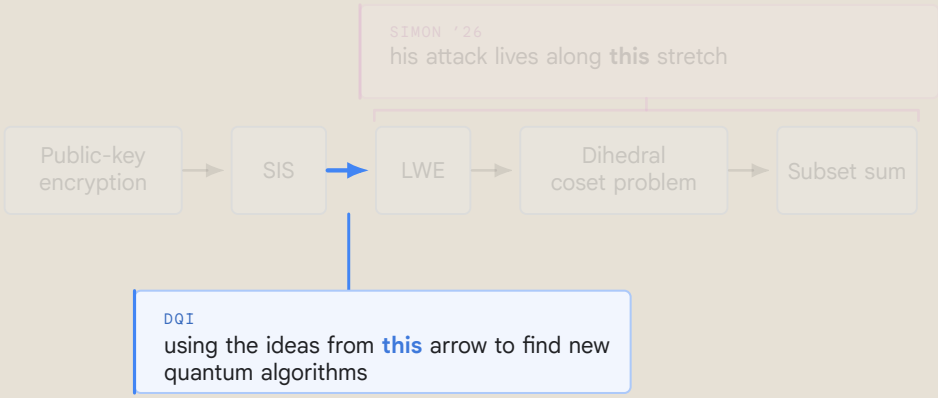
SIS reduces to LWE.

Generalisation: quantum decoding

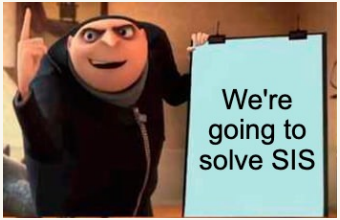
THE PROBLEM

given $\sum_{\mathbf{e}} \hat{g}(\mathbf{e})\,|A^{\mathsf{T}}\mathbf{s} + \mathbf{e}\rangle$, recover $A^{\mathsf{T}}\mathbf{s}$

Part 5: DQI — new algorithms via Regev's '05 reduction



Wait a minute...



Wait a minute...



Wait a minute...



Wait a minute...

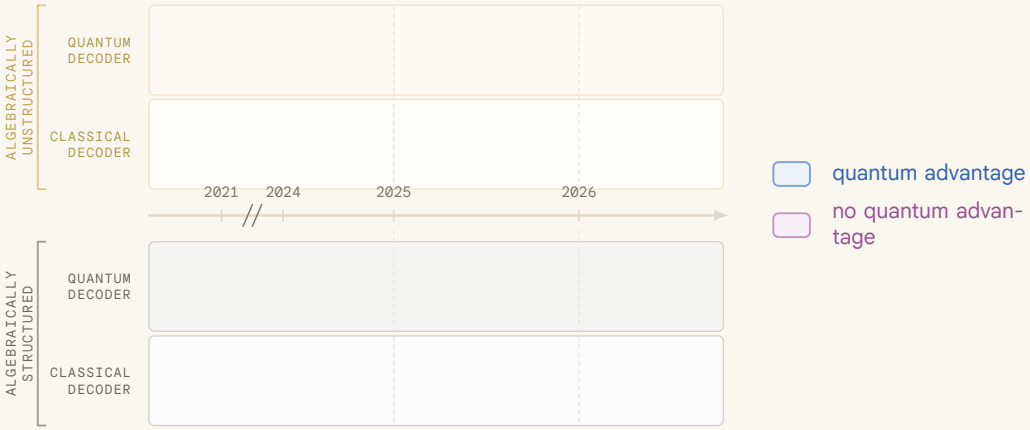


Wait a minute...

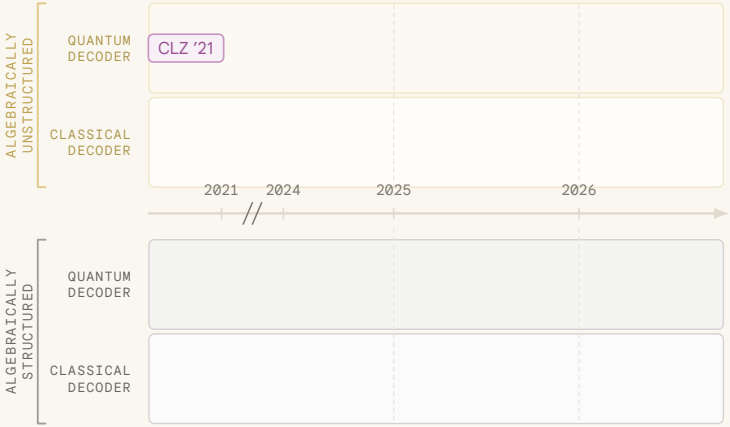


Keep the Regev '05 template, but find decoding problems that we can actually solve.

A history of DQI



A history of DQI

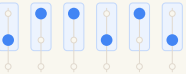


quantum advantage

no quantum advantage

$\mathbb{Z}_3$-SIS

$q = 3$, and $S_i = \{1, 2\}$ for every i

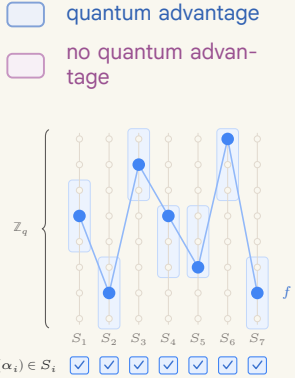


$\{1, 2\} = \mathbb{Z}_3 \setminus \{0\}$: no coordinate is allowed to vanish

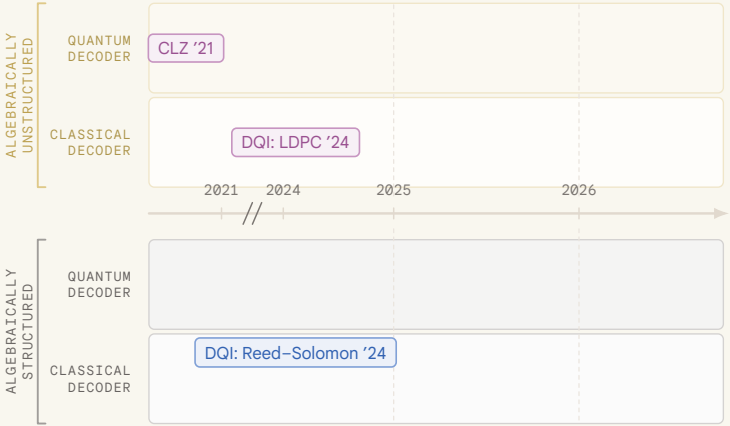
$|S_i|/q = 2/3$ — most of the alphabet is legal

A history of DQI





A history of DQI



quantum advantage

no quantum advantage

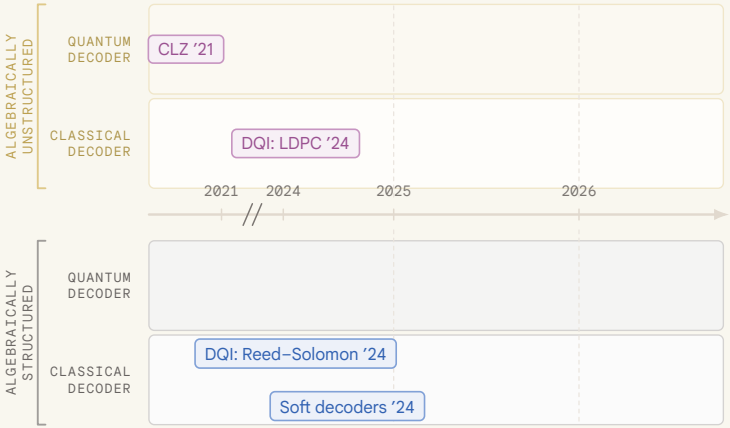
MAX-k-XORSAT



$\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i \bmod 2$

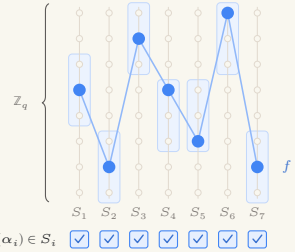
a few variables per constraint
— satisfy as many as you can

A history of DQI



quantum advantage

no quantum advantage

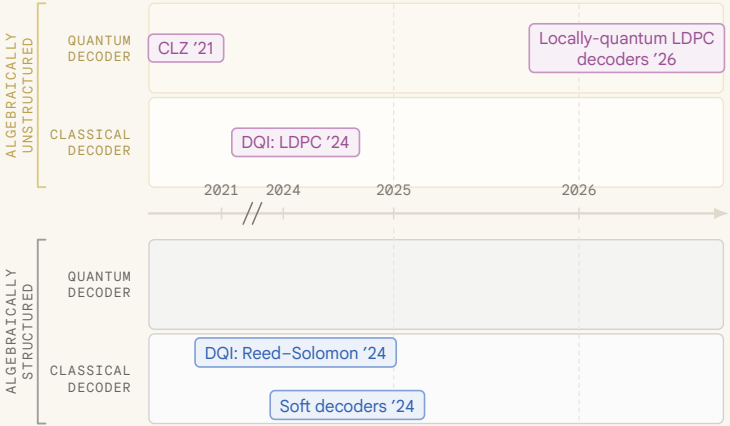


every coordinate, not just most of them

$$S = S_1 \times S_2 \times \cdots \times S_m$$

find a low-degree f that interpolates S

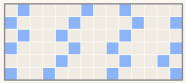
A history of DQI



quantum advantage

no quantum advantage

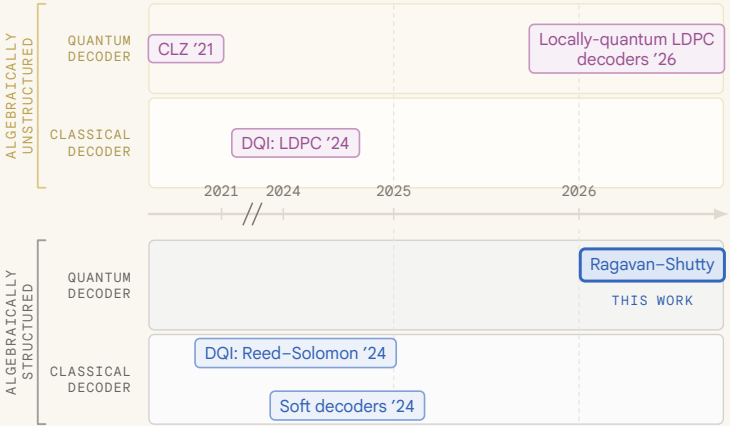
MAX-k-XORSAT



$\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i \bmod 2$

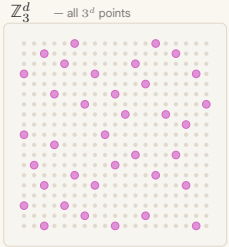
a few variables per constraint
— satisfy as many as you can

A history of DQI



quantum advantage

no quantum advantage

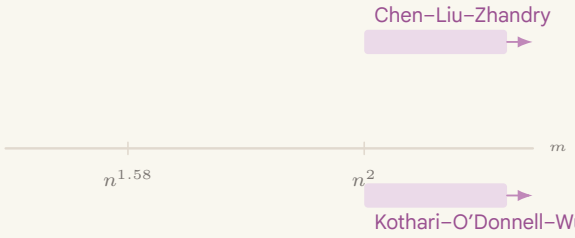


find a low-degree f that is nonzero
at only the pink points

Our observation: what CLZ actually needs

WHAT WE THOUGHT

$$m \geq n^2$$



Our observation: what CLZ actually needs

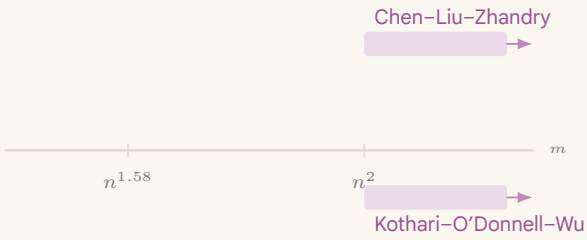
WHAT WE THOUGHT

~~$m \geq n^2$~~

WHAT IT ACTUALLY NEEDS

$m \geq \mathsf{rank}(C^{*2})$

C^{*2}, the Schur square: take every codeword of $C = \mathrm{im}\, A^{\mathsf{T}}$, square it entry by entry, and ask what those vectors span.



Our observation: what CLZ actually needs

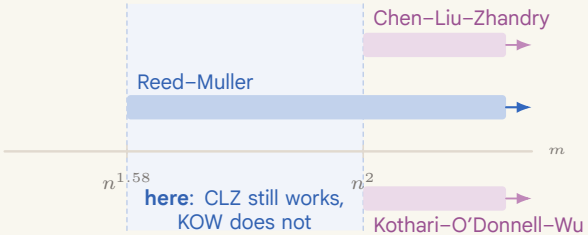
WHAT WE THOUGHT

~~$m \geq n^2$~~

WHAT IT ACTUALLY NEEDS

$m \geq \mathrm{rank}(C^{*2})$

C^{*2}, the Schur square: take every codeword of $C = \mathrm{im}\, A^{\mathsf{T}}$, square it entry by entry, and ask what those vectors span.



Our observation: what CLZ actually needs

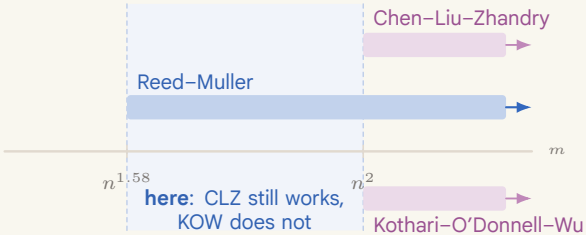
WHAT WE THOUGHT

~~$m \geq n^2$~~

WHAT IT ACTUALLY NEEDS

$m \geq \mathrm{rank}(C^{*2})$

C^{*2}, the Schur square: take every codeword of $C = \mathrm{im}\, A^{\mathsf{T}}$, square it entry by entry, and ask what those vectors span.



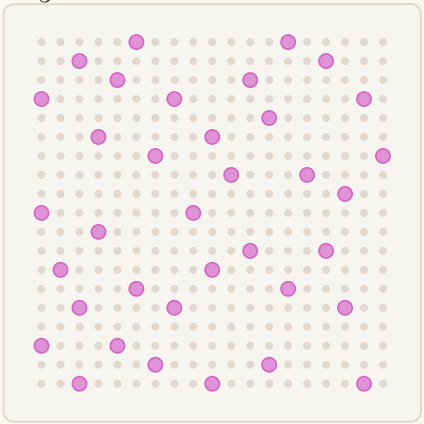
So why isn't this just OPI again? Because this problem really does need a quantum decoder.

An example problem that we solve

$\mathbb{Z}_3^d$ — all 3^d points

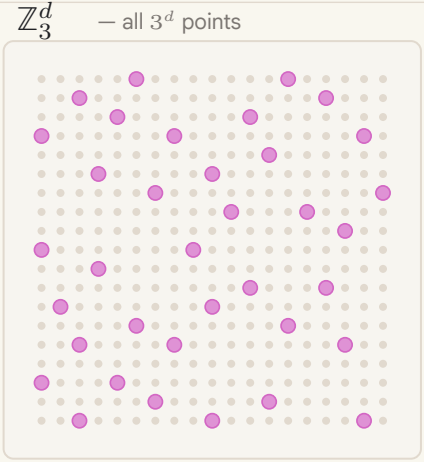
An example problem that we solve

$\mathbb{Z}_3^d$ — all 3^d points



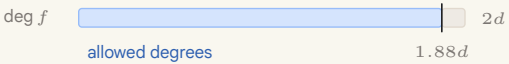
- Sample a set $S \subseteq \mathbb{Z}_3^d$ uniformly at random, of size $n := |S| = 3^{0.6\,d}$.

An example problem that we solve



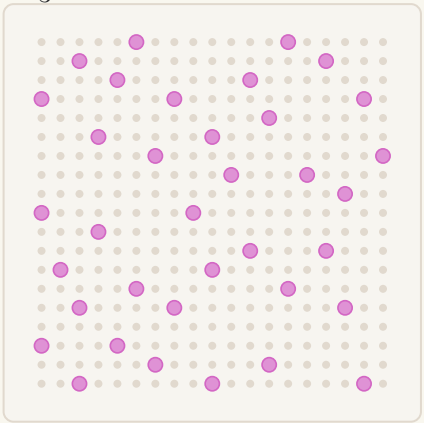
- Sample a set $S \subseteq \mathbb{Z}_3^d$ uniformly at random, of size $n := |S| = 3^{0.6\,d}$.
- Find $f \in \mathbb{Z}_3[X_1, \ldots, X_d]$ of degree $\deg f \leq 1.88\,d$ supported on exactly S:

$$f(z) = 0 \;\; (z \notin S), \qquad f(x) \neq 0 \;\; (x \in S).$$



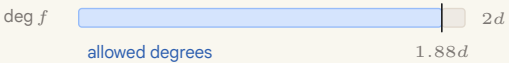
An example problem that we solve

$\mathbb{Z}_3^d$ — all 3^d points



- Sample a set $S \subseteq \mathbb{Z}_3^d$ uniformly at random, of size $n := |S| = 3^{0.6\,d}$.
- Find $f \in \mathbb{Z}_3[X_1, \ldots, X_d]$ of degree $\deg f \leq 1.88\,d$ supported on exactly S:

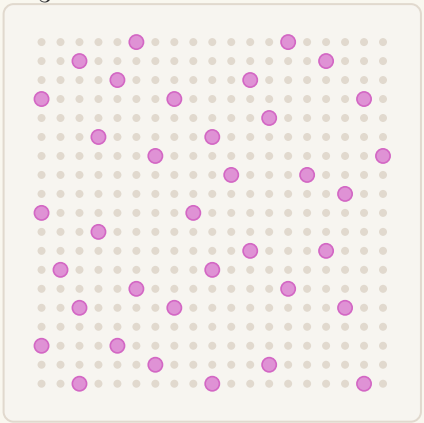
$$f(z) = 0 \;\; (z \notin S), \qquad f(x) \neq 0 \;\; (x \in S).$$



✓ quantum: **poly**(n) **time (our work)**

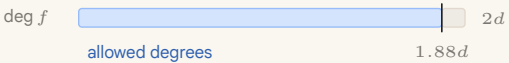
An example problem that we solve

$\mathbb{Z}_3^d$ — all 3^d points



- Sample a set $S \subseteq \mathbb{Z}_3^d$ uniformly at random, of size $n := |S| = 3^{0.6\,d}$.
- Find $f \in \mathbb{Z}_3[X_1, \ldots, X_d]$ of degree $\deg f \leq 1.88\,d$ supported on exactly S:

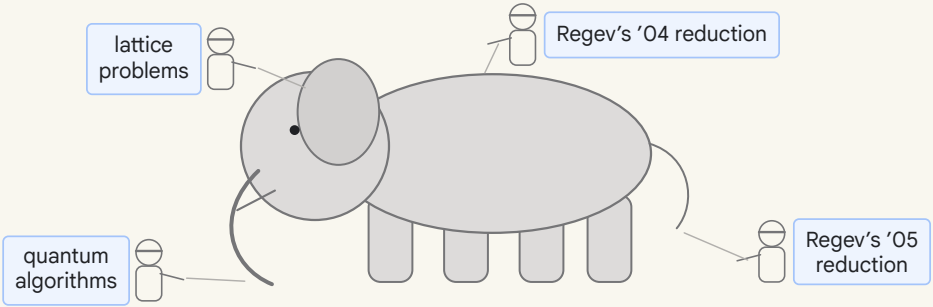
$$f(z) = 0 \ \ (z \notin S), \qquad f(x) \neq 0 \ \ (x \in S).$$



✓ quantum: **poly(n) time (our work)**

✗ classical: no poly(n)-time algorithm — SO FAR

Outlook


lattice problems
Regev's '04 reduction
quantum algorithms
Regev's '05 reduction

Thank you! Questions?

