Quantum Augmented Dual Attack

Martin R. Albrecht and Yixin Shen

Royal Holloway, University of London

29 November 2022

https://eprint.iacr.org/2022/656
Disclaimer

The submitted/eprint version contains a bug in the implementation of the estimator.

This presentation contains the corrected estimates which are worse than before but still better than the state of the art.

We thank Alessandro Budroni and Erik Mårtensson for pointing out this error to us.
Learning with errors (LWE)

Let $n = 4$, $m = 6$ and $q = 17$.

Given $A$ and $b$, find $s$. 

$$A \in \mathbb{Z}_q^{m \times n}$$

$$s \in \mathbb{Z}_q^n$$

$$b \in \mathbb{Z}_q^m$$

$$\begin{array}{cccc}
14 & 12 & 2 & 5 \\
5 & 3 & 1 & 7 \\
14 & 7 & 2 & 5 \\
0 & 9 & 8 & 4 \\
8 & 11 & 5 & 12 \\
5 & 1 & 3 & 14 \\
\end{array} \times \begin{array}{c}
\text{secret} \\
\text{secret} \\
\text{secret} \\
\text{secret} \\
\text{secret} \\
\text{secret} \\
\end{array} = \begin{array}{c}
11 \\
5 \\
14 \\
6 \\
12 \\
13 \\
\end{array}$$
Let $n = 4$, $m = 6$ and $q = 17$.

$$A \in \mathbb{Z}_q^{m \times n} \quad s \in \mathbb{Z}_q^n \quad b \in \mathbb{Z}_q^m$$

$$\begin{bmatrix} 14 & 12 & 2 & 5 \\ 5 & 3 & 1 & 7 \\ 14 & 7 & 2 & 5 \\ 0 & 9 & 8 & 4 \\ 8 & 11 & 5 & 12 \\ 5 & 1 & 3 & 14 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \\ 14 \\ 6 \\ 12 \\ 13 \end{bmatrix}$$

Given $A$ and $b$, find $s$.

~ Very easy (e.g. Gaussian elimination) and in polynomial time
Learning with errors (LWE)

Let $n = 4$, $m = 6$ and $q = 17$.

| random $A \in \mathbb{Z}_q^{m \times n}$ | secret $s \in \mathbb{Z}_q^n$ | noise $e \in \mathbb{Z}_q^m$ | $b \in \mathbb{Z}_q^m$ |
|----------------------------------------|-------------------------------|------------------------------|------------------------|
| 14 12 2 5                             | 1                             | -3                           | 11                     |
| 5 3 1 7                               | 2                             | -1                           | 5                      |
| 14 7 2 5                              | 1                             | 2                            | 14                     |
| 0 9 8 4                               | 5                             | -3                           | 6                      |
| 8 11 5 12                             |                                | 3                            | 12                     |
| 5 1 3 14                              |                                | -1                           | 13                     |

Given $A$ and $b$, find $s$. 

\[ A \times s + e = b \]
Learning with errors (LWE)

Let \( n = 4 \), \( m = 6 \) and \( q = 17 \).

\[
\begin{align*}
\text{random} & : \quad A \in \mathbb{Z}_q^{m \times n} \\
\text{secret} & : \quad s \in \mathbb{Z}_q^n \\
\text{noise} & : \quad e \in \mathbb{Z}_q^m \\
\text{solution} & : \quad b \in \mathbb{Z}_q^m
\end{align*}
\]

\[
\begin{bmatrix}
14 & 12 & 2 & 5 \\
5 & 3 & 1 & 7 \\
14 & 7 & 2 & 5 \\
0 & 9 & 8 & 4 \\
8 & 11 & 5 & 12 \\
5 & 1 & 3 & 14
\end{bmatrix}
\times
\begin{bmatrix}
\text{red} \\
\text{red} \\
\text{red} \\
\text{red} \\
\text{red} \\
\text{red}
\end{bmatrix}
\quad +
\begin{bmatrix}
\text{green} \\
\text{green} \\
\text{green} \\
\text{green} \\
\text{green} \\
\text{green}
\end{bmatrix}
= 
\begin{bmatrix}
11 \\
5 \\
14 \\
6 \\
12 \\
13
\end{bmatrix}
\]

Given \( A \) and \( b \), find \( s \).

\( \sim \) Suspected hard problem, even for quantum algorithms
Learning with errors (LWE)

\[ \text{LWE}(n, m, q, \chi_e, \chi_s) : \text{probability distribution on } \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \]

- sample \( A \leftarrow U(\mathbb{Z}_q^{m \times n}) \)
- sample \( s \leftarrow \chi_s^n \)
- sample \( e \leftarrow \chi_e^m \)
- output \( (A, As + e) \).
Learning with errors (LWE)

$LWE(n, m, q, \chi_e, \chi_s)$: probability distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$

- sample $A \leftarrow U(\mathbb{Z}_q^{m \times n})$
- sample $s \leftarrow \chi_s^n$
- sample $e \leftarrow \chi_e^m$
- output $(A, As + e)$.

Secret distributions $\chi_s$:
- originally uniform in $\mathbb{Z}_q$, now some distribution of small deviation $\sigma_s$ (e.g. discrete Gaussian/centered Binormal, $\{-1, 0, 1\}$ whp)
- Fact: small secret is as hard as uniform secret
- small secret allows more efficient schemes

Noise distributions $\chi_e$:
- usually discrete Gaussian/centered Binormal of deviation $\sigma_e$
- most schemes (Kyber/Saber/...): $\sigma_e$ small ($\approx 1$)
LWE: security and attacks

LWE is fundamental to lattice-based cryptography:

- several lattice-based NIST PQC candidates rely on LWE
- extensive literature
- all evidence points to resistance against quantum attacks

Two types of attacks:

- Primal attacks:
  - more efficient
  - no quantum speed-up known (besides BKZ)

- Dual attacks:
  - originally less efficient, now catching up
  - no quantum speed-up known (besides BKZ) up to now

Contribution: first quantum speed-up on dual attacks
LWE: security and attacks

LWE is fundamental to lattice-based cryptography:

► several lattice-based NIST PQC candidates rely on LWE
► extensive literature
► all evidence points to resistance against quantum attacks

Two types of attacks:

► Primal attacks:
  ► more efficient
  ► no quantum speed-up known (besides BKZ)
► Dual attacks:
  ► originally less efficient, now catching up
  ► no quantum speed-up known (besides BKZ) up to now

Contribution: first quantum speed-up on dual attacks
Modern dual attacks

Many techniques used to obtain improvements:

- hybrid attacks: guess part of the secret exhaustively
- modulo switching to reduce modulo $q$
- BKZ with sieving to produce many dual vectors at once
- sophisticated statistical analysis

~ [MAT22] contains all the details
Modern dual attacks

Many techniques used to obtain improvements:

- hybrid attacks: guess part of the secret exhaustively
- modulo switching to reduce modulo $q$
- BKZ with sieving to produce many dual vectors at once
- sophisticated statistical analysis

$\Rightarrow$ [MAT22] contains all the details

But fundamentally reduces the problem to distinguishing a uniform distribution from a modular discrete Gaussian

$\Rightarrow$ compute Fourier transform
Given a sampler for $\chi$, decide if $\chi = U(\mathbb{Z}_q)$ or $D_{\sigma,q}$ (discrete Gaussian)
Uniform/Gaussian distinguisher

Given a sampler for $\chi$, decide if $\chi = U(\mathbb{Z}_q)$ or $D_{\sigma,q}$ (discrete Gaussian)

Essentially optimal distinguisher: use Fourier transform

$$\mathbb{E}_{x \leftarrow \chi}[e^{2i\pi x/q}], \text{Var}_{x \leftarrow \chi}[e^{2i\pi x/q}] \approx \begin{cases} 0, 0 & \text{if } \chi = U(\mathbb{Z}_q) \\ e^{-2\left(\frac{\pi \sigma}{q}\right)^2}, e^{-8\left(\frac{\pi \sigma}{q}\right)^2} & \text{if } \chi = D_{\sigma,q} \end{cases}$$
Uniform/Gaussian distinguisher

Given a sampler for \( \chi \), decide if \( \chi = U(\mathbb{Z}_q) \) or \( D_{\sigma,q} \) (discrete Gaussian)

Essentially optimal distinguisher: use Fourier transform

\[
\mathbb{E}_{x \leftarrow \chi} [e^{2i\pi x/q}], \quad \text{Var}_{x \leftarrow \chi} [e^{2i\pi x/q}] \approx \begin{cases} 
0, 0 & \text{if } \chi = U(\mathbb{Z}_q) \\
\ e^{-2\left(\frac{\pi \sigma}{q}\right)^2}, \ e^{-8\left(\frac{\pi \sigma}{q}\right)^2} & \text{if } \chi = D_{\sigma,q}
\end{cases}
\]

Attack:

- sample \( N = \Omega\left(1/\varepsilon^2\right) \) values \( x_1, \ldots, x_N \) from \( \chi \)
- compute

\[
S = \frac{1}{N} \sum_{j=1}^{N} e^{2i\pi x_j/q}
\]

- Check if \( S > e^{-2\left(\frac{\pi \sigma}{q}\right)^2} \)

The quantity \( \varepsilon = e^{-2\left(\frac{\pi \sigma}{q}\right)^2} \) is called the advantage.
Modern dual attack at the high-level

All you need to know for what follows: attack looks like

- enumerate \( s_{\text{enum}} \in \mathbb{Z}_q^{k_{\text{enum}}} \)
  - enumerate all \( s_{\text{fft}} \in \mathbb{Z}_q^{k_{\text{fft}}} \)
    - compute an DFT-like sum
    - check if it is above the threshold

Classical complexity:
\[
G(\chi_{\text{s}}^{k_{\text{enum}}} s) \cdot q^{k_{\text{fft}}} + N, \quad N = \text{# of samples to distinguish}
\]

Guessing complexity: try \( s_{\text{enum}} \) in decreasing order of probability

FFT: compute all DFT-sums in one go with an FFT \cite{GJ21}

Quantum complexity: hope for
\[
q G(\chi_{\text{s}}^{k_{\text{enum}}} s) \cdot q^{k_{\text{fft}}} + N
\]

Unclear
Modern dual attack at the high-level

All you need to know for what follows: attack looks like

- enumerate \( s_{\text{enum}} \in \mathbb{Z}_q^{k_{\text{enum}}} \)
- enumerate all \( s_{\text{fft}} \in \mathbb{Z}_q^{k_{\text{fft}}} \)
  - compute an DFT-like sum
  - check if it is above the threshold

Classical complexity:

\[
G(\chi_s^{k_{\text{enum}}}) \cdot (q^{k_{\text{fft}}} + N), \quad N = \# \text{ of samples to distinguish}
\]

- guessing complexity: try \( s_{\text{enum}} \) in decreasing order of probability
- FFT: compute all DFT-sums in one go with an FFT [GJ21]
Modern dual attack at the high-level

All you need to know for what follows: attack looks like

- enumerate \( s_{\text{enum}} \in \mathbb{Z}_q^{k_{\text{enum}}} \)
- enumerate all \( s_{\text{fft}} \in \mathbb{Z}_q^{k_{\text{fft}}} \)
  - compute an DFT-like sum
  - check if it is above the threshold

Classical complexity:

\[
G(\chi_s^{k_{\text{enum}}}) \cdot (q^{k_{\text{fft}}} + N), \quad N = \# \text{ of samples to distinguish}
\]

- guessing complexity: try \( s_{\text{enum}} \) in decreasing order of probability
- FFT: compute all DFT-sums in one go with an FFT [GJ21]

Quantum complexity: hope for \( \sqrt{G(\chi_s^{k_{\text{enum}}}) \cdot (q^{k_{\text{fft}}} + N)} \)?
Modern dual attack at the high-level

All you need to know for what follows: attack looks like

- enumerate \( s_{\text{enum}} \in \mathbb{Z}_q^{k_{\text{enum}}} \)
- enumerate all \( s_{\text{fft}} \in \mathbb{Z}_q^{k_{\text{fft}}} \)
  - compute an DFT-like sum
  - check if it is above the threshold

Classical complexity:

\[
G(\chi_s^{k_{\text{enum}}}) \cdot (q^{k_{\text{fft}}} + N), \quad N = \text{# of samples to distinguish}
\]

- guessing complexity: try \( s_{\text{enum}} \) in decreasing order of probability
- FFT: compute all DFT-sums in one go with an FFT [GJ21]

Quantum complexity: hope for \( \sqrt{G(\chi_s^{k_{\text{enum}}}) \cdot (q^{k_{\text{fft}}} + N)} \) ? Unclear
Guessing complexity

$D$ discrete distribution on $x_1, x_2, \ldots$, let $p_i$ be the probability of $x_i$.

**Guessing game:** your friend secretly samples $X \leftarrow D$, you must find $i$ such that $X = x_i$ only by asking queries of the form “is $X = x_j$?” for some $j$. Minimize (expected) number of queries.
Guessing complexity

$D$ discrete distribution on $x_1, x_2, \ldots$, let $p_i$ be the probability of $x_i$.

**Guessing game:** your friend secretly samples $X \leftarrow D$, you must find $i$ such that $X = x_i$ only by asking queries of the form “is $X = x_j$?” for some $j$. Minimize (expected) number of queries.

**Optimal strategy:** always guess elements by decreasing probability

Expected number of guesses ($p_1 \geq p_2 \geq \cdots \geq p_N$):

$$G(D) = \sum_{i=1}^{N} i \cdot p_i,$$
Guessing complexity

$D$ discrete distribution on $x_1, x_2, \ldots$, let $p_i$ be the probability of $x_i$.

**Guessing game:** your friend secretly samples $X \leftarrow D$, you must find $i$ such that $X = x_i$ only by asking queries of the form “is $X = x_j$?” for some $j$. Minimize (expected) number of queries.

**Optimal strategy:** always guess elements by decreasing probability

Expected number of guesses ($p_1 \geq p_2 \geq \cdots \geq p_N$):

$$G(D) = \sum_{i=1}^{N} i \cdot p_i, \quad G^{qc}(D) = \sum_{i=1}^{N} \sqrt{i} \cdot p_i$$

What about quantum guessing?

- Grover-like search [Mon10]
- can even handle faulty query oracles (our contribution)
Guessing complexity (results)

Guessing complexity of the modular discrete Gaussian $D_{\sigma,q,n}$ on $\mathbb{Z}_q^n$:

$$D_{\sigma,q,n}(x) \propto \rho_\sigma(x + q \mathbb{Z}^n), \quad \rho_\sigma(y) = e^{-\|y\|^2/2\sigma}, \quad y \in \mathbb{Z}^n.$$
Guessing complexity (results)

Guessing complexity of the **modular discrete Gaussian** $D_{\sigma,q,n}$ on $\mathbb{Z}_q^n$:

$$D_{\sigma,q,n}(x) \propto \rho_{\sigma}(x + q \mathbb{Z}^n), \quad \rho_{\sigma}(y) = e^{-\|y\|^2/2\sigma}, \quad y \in \mathbb{Z}^n.$$  

**Theorem (Simplified)**

$$G(D_{\sigma,q,n}) \lesssim 1.22^n \cdot 2^H, \quad G^{qc}(D_{\sigma,q,n}) \lesssim 1.12^{n/2} \cdot 2^{H/2}$$

where $H \approx \frac{1/2 + \log(\sigma \sqrt{2\pi})}{\log 2}$ is the entropy of the discrete Gaussian.

**Observations:**

- $G$ exponentially times bigger than $2^H$
- $G^{qc} \leq \sqrt{G}$ is true for any distribution
- $G^{qc}$ seems exponentially smaller than $\sqrt{G}$ ...
- ... but we do not have matching lower bounds to confirm it yet
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{fft}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

\[\text{find } s \in \mathbb{Z}_q^{k_{fft}} \text{ s.t. } |F(s)| > \delta \text{ where } F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\]
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{\text{fft}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

\[\text{find } s \in \mathbb{Z}_q^{\text{fft}} \text{ s.t. } |F(s)| > \delta\] where

\[F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\]

Naive complexity:

\[O(q^{\text{fft}} \cdot N)\]
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}^k_{q \text{fft}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}^k_{q \text{fft}}\) s.t. \(|F(s)| > \delta\) where 
  \[
  F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}
  \]

Naive complexity:

\[
O(q^{k_{\text{fft}}} \cdot N)
\]

Classical algorithm with optimisation: [GJ21]

- \(T \leftarrow k_{\text{fft}}\)-dimensional array set to zero
- \(T[x_j] \leftarrow w_j\) for all \(j\)
- compute FFT \(\hat{T}\) of \(T\) (Fact: \(\hat{T}[s] = F(s)\))
- check all \(\hat{T}[s]\) against threshold
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^N w_j \cdot e^{-2i\pi s^T x_j / q}\)

Naive complexity:

\[O(q^{k_{\text{fft}}} \cdot N)\]

Classical algorithm with optimisation: \([GJ21]\)

- \(T \leftarrow k_{\text{fft}}\)-dimensional array set to zero
- \(T[x_j] \leftarrow w_j\) for all \(j\)
- compute FFT \(\hat{T}\) of \(T\) (Fact: \(\hat{T}[s] = F(s)\))
- check all \(\hat{T}[s]\) against threshold

Complexity:

\[\text{array filling time} + \text{FFT time} + \text{search time} = O(N + q^{k_{\text{fft}}})\]
FFT search with threshold

Problem: given $(x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}$ with $N$ large and $\delta > 0$

► find $s \in \mathbb{Z}_q^{k_{\text{fft}}}$ s.t. $|F(s)| > \delta$ where $F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}$

What about quantum? initial idea: use the QFT
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{fft}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_q^{k_{fft}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j/q}\)

What about quantum? initial idea: use the QFT

- create superposition

\[
\psi = \frac{1}{Z} \sum_{j=1}^{N} w_j |x_j\rangle
\]
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^k \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

\[\text{find } s \in \mathbb{Z}_q^k \text{ s.t. } |F(s)| > \delta \text{ where } F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2\pi i s^T x_j / q}\]

What about quantum? initial idea: use the QFT

\[\psi = \frac{1}{Z} \sum_{j=1}^{N} w_j |x_j\rangle\]

apply QFT to get

\[\hat{\psi} = \frac{1}{Z} \sum_{s \in \mathbb{Z}_q^k} F(s) |s\rangle\]
FFT search with threshold

**Problem:** given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_{q}^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_{q}^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2\pi i s^T x_j / q}\)

What about quantum? initial idea: use the QFT

- create superposition

\[
\psi = \frac{1}{\mathcal{Z}} \sum_{j=1}^{N} w_j |x_j\rangle
\]

- apply QFT to get

\[
\hat{\psi} = \frac{1}{\mathcal{Z}} \sum_{s \in \mathbb{Z}_{q}^{k}} F(s) |s\rangle
\]

- check if any amplitude in the superposition is above the threshold
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

▶ find \(s \in \mathbb{Z}^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\)

What about quantum? initial idea: use the QFT

▶ create superposition

\[
\psi = \frac{1}{Z} \sum_{j=1}^{N} w_j |x_j\rangle
\]

▶ apply QFT to get

\[
\hat{\psi} = \frac{1}{Z} \sum_{s \in \mathbb{Z}_q^k} F(s) |s\rangle
\]

▶ check if any amplitude in the superposition is above the threshold
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2\pi i s^T x_j / q}\)

What about quantum? initial idea: use the QFT

- create superposition
  \[
  \psi = \frac{1}{Z} \sum_{j=1}^{N} w_j |x_j\rangle
  \]

- apply QFT to get
  \[
  \hat{\psi} = \frac{1}{Z} \sum_{s \in \mathbb{Z}_q^k} F(s) |s\rangle
  \]

- check if any amplitude in the superposition is above the threshold

- impossible without QRAM?
- polynomial time

Open question: can this approach be made efficient?
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- Find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\)

What about quantum? Initial idea: use the QFT

- Create superposition

\[
\psi = \frac{1}{Z} \sum_{j=1}^{N} w_j \left| x_j \right> 
\]

- Apply QFT to get

\[
\hat{\psi} = \frac{1}{Z} \sum_{s \in \mathbb{Z}_q^{k}} F(s) \left| s \right>
\]

- Check if any amplitude in the superposition is above the threshold

Open question: can this approach be made efficient?
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where
  \[
  F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}
  \]

Alternative quantum algorithm:

- search over \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) with Grover
  - compute \(F(s)\) and check against threshold

Theorem (Simplified)

There is a quantum algorithm that computes \(F(s) \pm \varepsilon\) given oracle access by making \(O\left(\frac{1}{\varepsilon}\right)\) queries to \(O_X: |j\rangle|0\rangle \rightarrow |j\rangle x_j\).

How can we build such an oracle?
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)
► find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\)

Alternative quantum algorithm:
► search over \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) with Grover
  ► compute \(F(s)\) and check against threshold

Complexity: \(O(\sqrt{q^{k_{\text{fft}}} \cdot N})\) ► worse than classical unless \(N < \sqrt{q^{k_{\text{fft}}}}\)
FFT search with threshold

Problem: given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- find \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) s.t. \(|F(s)| > \delta\) where \(F(s) = \sum_{j=1}^{N} w_j \cdot e^{-2i\pi s^T x_j / q}\)

Alternative quantum algorithm:

- search over \(s \in \mathbb{Z}_q^{k_{\text{fft}}}\) with Grover
  - compute \(F(s)\) and check against threshold

Complexity: \(O(\sqrt{q^{k_{\text{fft}}} \cdot N})\) - worse than classical unless \(N < \sqrt{q^{k_{\text{fft}}}}\)

- we can do better when \(N > \sqrt{q^{k_{\text{fft}}}}\) with a QRAM

Theorem (Simplified)

There is a quantum algorithm that computes \(F(s) \pm \varepsilon\) given oracle access by making \(O(1/\varepsilon)\) queries to \(O_X:\)

\[ O_X : |j\rangle |0\rangle \rightarrow |j\rangle |x_j\rangle. \]

How can we build such an oracle? \(\sim\) QRAM
FFT search with threshold (quantum)

Given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_{q}^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- put \((x_j, w_j)\) in a QRAM \(\mathcal{O}_X\)
- search over \(s \in \mathbb{Z}^k_q\) with Grover
  - compute \(F(s)\) using theorem with \(\mathcal{O}_X\) and check against threshold \(\delta\)

Theorem (Simplified)

*There is a quantum algorithm that computes \(F(s) \pm \varepsilon\) given oracle access by making \(O(1/\varepsilon)\) queries to \(\mathcal{O}_X\).*
FFT search with threshold (quantum)

Given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- put \((x_j, w_j)\) in a QRAM \(\mathcal{O}_X\)
- search over \(s \in \mathbb{Z}_q^k\) with Grover
  - compute \(F(s)\) using theorem with \(\mathcal{O}_X\) and check against threshold \(\delta\)

Theorem (Simplified)

*There is a quantum algorithm that computes \(F(s) \pm \varepsilon\) given oracle access by making \(O(1/\varepsilon)\) queries to \(\mathcal{O}_X\).*

What about \(\varepsilon\)?
FFT search with threshold (quantum)

Given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{fft}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

\[\uparrow \quad \text{put } (x_j, w_j) \text{ in a QRAM } \mathcal{O}_X\]

\[\uparrow \quad \text{search over } s \in \mathbb{Z}_q^k \text{ with Grover}\]

\[\uparrow \quad \text{compute } F(s) \text{ using theorem with } \mathcal{O}_X \text{ and check against threshold } \delta\]

Theorem (Simplified)

*There is a quantum algorithm that computes* \(F(s) \pm \varepsilon\) *given oracle access by making* \(O(1/\varepsilon)\) *queries to* \(\mathcal{O}_X\).*

What about \(\varepsilon\)? For dual attacks: \(\varepsilon = \Omega(1/\sqrt{N})\)

Quantum complexity

\[O(\sqrt{q^{k_{fft}} \cdot N})\]
FFT search with threshold (quantum)

Given \((x_1, w_1), \ldots, (x_N, w_N) \in \mathbb{Z}_q^{k_{\text{fft}}} \times \mathbb{C}\) with \(N\) large and \(\delta > 0\)

- put \((x_j, w_j)\) in a QRAM \(\mathcal{O}_X\)
- search over \(s \in \mathbb{Z}_q^k\) with Grover
  - compute \(F(s)\) using theorem with \(\mathcal{O}_X\) and check against threshold \(\delta\)

**Theorem (Simplified)**

*There is a quantum algorithm that computes \(F(s) \pm \varepsilon\) given oracle access by making \(O(1/\varepsilon)\) queries to \(\mathcal{O}_X\).*

What about \(\varepsilon\)? For dual attacks: \(\varepsilon = \Omega(1/\sqrt{N})\)

**Quantum complexity**

\[O(\sqrt{q^{k_{\text{fft}}} \cdot N})\]

- quantum never worse than classical
- gain when \(N \ll q^{k_{\text{fft}}}\): like in dual attacks
## Dual attack cost estimates (logarithms to base two)

| Scheme       | Classical | Quantum | Our work |
|--------------|-----------|---------|----------|
|              | CC        | CN      | C0       | QN | Q0 | QN | Q0 |
| Kyber 512    | 139.2     | 134.4   | 115.4    | 124.4 | 102.7 | 119.3 | 99.6 |
| Kyber 768    | 196.1     | 190.6   | 173.7    | 175.3 | 154.6 | 168.2 | 149.8 |
| Kyber 1024   | 262.4     | 256.1   | 241.8    | 234.5 | 215.0 | 226.0 | 208.5 |
| LightSaber   | 138.5     | 133.1   | 113.7    | 122.7 | 101.1 | 118.6 | 98.5  |
| Saber        | 201.4     | 195.9   | 179.2    | 179.9 | 159.4 | 175.6 | 155.7 |
| FireSaber    | 263.5     | 258.2   | 243.8    | 235.9 | 216.7 | 228.3 | 210.7 |
| TFHE630      | 118.2     | 113.3   | 93.0     | 105.2 | 83.0  | 102.6 | 81.6  |
| TFHE1024     | 122.0     | 117.2   | 95.4     | 108.5 | 84.8  | 106.6 | 83.5  |

- **QN**: quantum version of CN
- **Q0**: quantum version of C0
- **CC**: classical circuit model (most detailed)
- **CN**: classical query model (intermediate)
- **C0**: Core-SVP model (very pessimistic)
Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to crystals. In *Advances in Cryptology – ASIACRYPT 2021*, pages 33–62, Cham, 2021. Springer International Publishing.

MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack. Available at https://doi.org/10.5281/zenodo.6412487, April 2022.

Ashley Montanaro. Quantum search with advice. In *Theory of Quantum Computation, Communication, and Cryptography TQC 2010*, 2010.