Time and Query Complexity Tradeoffs for the Dihedral Coset Problem

Maxime Remaud[1], **André Schrottenloher**[2], Jean-Pierre Tillich[3]

[1] Inria and Eviden Quantum Lab
[2] Univ Rennes, Inria, CNRS, IRISA
[3] Inria

Abelian hidden shift & DCP

Abelian hidden shift

Given an abelian group $(G, +)$, **two functions** $f, g \ : G \to S$ such that: $\exists \mathbf{s}, f(x) = g(x + \mathbf{s})$, find $\mathbf{s}$.

$\implies$ underlies the security of some post-quantum schemes (e.g. CSIDH)

$G = \mathbb{Z}_{\mathbf{N}}$ in this talk.

Abelian hidden shift & DCP (ctd.)

Quantum 101:

- operate on states $\sum_x \alpha_x |x\rangle$ with **complex amplitudes** α_x
- α_x is not observable, measuring returns x with probability $|\alpha_x|^2$

- Create a superposition:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{\mathbf{N}}}\sum_{x\in\mathbb{Z}_{\mathbf{N}}}|x\rangle$$

- Apply f if 0, g if 1:

$$\frac{1}{\sqrt{2\mathbf{N}}}|0\rangle\sum_x |x\rangle|f(x)\rangle + \frac{1}{\sqrt{2\mathbf{N}}}|1\rangle\sum_x |x\rangle|g(x)\rangle$$

- Measure the output: get $\frac{1}{\sqrt{2}}(|0\rangle|x\rangle + |1\rangle|x+\mathbf{s}\rangle)$

DCP: Find $\mathbf{s}$ from such states (with random x).

Phase vectors

Let $\omega_N = \exp(2i\pi/N)$

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

$$\xrightarrow{QFT_{\mathbf{N}}} \frac{1}{\sqrt{2\mathbf{N}}} \sum_{k\in\mathbb{Z}_{\mathbf{N}}} \omega_{\mathbf{N}}^{kx} |k\rangle \left(|0\rangle + \omega_{\mathbf{N}}^{k\mathbf{s}} |1\rangle\right)$$

$$\xrightarrow{\text{Measure 1st register}} \frac{1}{\sqrt{2}} |k\rangle \left(|0\rangle + \omega_{\mathbf{N}}^{k\mathbf{s}} |1\rangle\right) =: |\psi_k\rangle$$

- From now on, **combine** random phase vectors $|\psi_k\rangle$ into **more useful ones** (some constraint on k)
- Ex. if $\mathbf{N}$ is even, $k = \mathbf{N}/2 \implies |\psi_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$, gives one bit of $\mathbf{s}$

Quantum algorithms for abelian hidden shift ($n = \log_2 N$)

- Ettinger-Høyer: $\mathcal{O}(\mathbf{n})$ queries and $\mathcal{O}(2^{\mathbf{n}})$ time
- Kuperberg (2003): $2^{\mathcal{O}(\sqrt{\mathbf{n}})}$ queries and time
- Regev: $2^{\mathcal{O}(\sqrt{\mathbf{n}\log\mathbf{n}})}$ queries and time
- Kuperberg (2011): $\mathcal{O}\left(2^{\sqrt{2\mathbf{n}}}\right)$ queries and time

. . . and many trade-offs to attack CSIDH instances, especially with **lower query complexity**.

This paper

- Improved algorithm with linear query complexity
- Interpolation between this algorithm and Kuperberg's (2011)

(Limited) impact on CSIDH

Reducing the amount of queries is important, but not enough: we also need to keep the time small.

$\Longrightarrow$ improving the attacks on CSIDH would require **better quantum algorithms for subset-sum**

Linear-queries Algorithm

Regev's combination routine

Idea:

- combine many $|\psi_{k_i}\rangle$ ($\mathbf{k} = (k_1, \ldots, k_m)$)
- create a new $|\psi_k\rangle$ with **a condition on** k (**such that** $B|k$)

1: Write:

$$|\psi_{k_1}\rangle\,|\psi_{k_2}\rangle \cdots |\psi_{k_m}\rangle = \frac{1}{\sqrt{2^m}} \sum_{\mathbf{b}\in\{0,1\}^m} \omega_{\mathbf{N}}^{\mathbf{s}\mathbf{b}\cdot\mathbf{k}}\,|\mathbf{b}\rangle$$

2: Compute $\mathbf{k}\cdot\mathbf{b} \mod B$ and measure the value z

3: **Compute the vectors** $\mathbf{b}$ such that $\mathbf{b}\cdot\mathbf{k} = z$

4: If there are two vectors, the state is:

$$\frac{1}{\sqrt{2}}\left(\omega_{\mathbf{N}}^{\mathbf{s}\mathbf{b_1}\cdot\mathbf{k}}\,|\mathbf{b}_1\rangle + \omega_{\mathbf{N}}^{\mathbf{s}\mathbf{b_2}\cdot\mathbf{k}}\,|\mathbf{b}_2\rangle\right) \underbrace{\simeq}_{\text{global phase}} \frac{1}{\sqrt{2}}\left(|\mathbf{b}_1\rangle + \omega_{\mathbf{N}}^{\mathbf{s}\overbrace{(\mathbf{b}_2-\mathbf{b}_1)\cdot\mathbf{k}}^{\text{New label } k}}\,|\mathbf{b}_2\rangle\right)$$

Regev, A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space (2004), arXiv:quant-ph/0406151

Consequences

- The difficult step: computing $\mathbf{b}_1$ and $\mathbf{b}_2$

$\Longrightarrow$ (random) subset-sum problem

given $\mathbf{k} = (k_1, \ldots, k_m)$, given z, find $\mathbf{b} \in \{0,1\}^m$ s.t. $\mathbf{b} \cdot \mathbf{k} = z \mod B$

- In Regev's algorithm, we use multiple levels with $m \simeq \log_2 B \simeq \sqrt{\mathbf{n}}$
  $\Longrightarrow \simeq \sqrt{\mathbf{n}}^{\sqrt{\mathbf{n}}}$ operations & queries
- We can also combine $\simeq \mathbf{n}$ phase vectors to recover one bit of $\mathbf{s}$, and repeat

$\Longrightarrow \simeq \mathbf{n}^2$ queries (good for attacking CSIDH-512)

Regev, A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space (2004), arXiv:quant-ph/0406151

New algorithm

Reduce the queries to $\mathcal{O}(\mathbf{n})$ but keep the time $\ll 2^{\mathbf{n}}$: recover the full secret in one pass.

1: As before, compute $\mathbf{k} \cdot \mathbf{b} \mod \mathbf{N}$

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{b} \in \{0,1\}^m} \omega_{\mathbf{N}}^{\mathbf{s}\mathbf{b}\cdot\mathbf{k}} |\mathbf{b}\rangle |\mathbf{k} \cdot \mathbf{b}\rangle$$

2: **Solve the subset-sum problem** to compute $\mathbf{b}$ from $\mathbf{k} \cdot \mathbf{b}$
3: **Remove b**

$$\frac{1}{\sqrt{2^m}} \sum_{\mathbf{b} \in \{0,1\}^m} \omega_{\mathbf{N}}^{\mathbf{s}\mathbf{b}\cdot\mathbf{k}} |\mathbf{k} \cdot \mathbf{b}\rangle \simeq \frac{1}{\sqrt{\mathbf{N}}} \sum_x \omega_N^{\mathbf{s}x} |x\rangle$$

4: This is the QFT of $|\mathbf{s}\rangle$, so apply the inverse QFT and measure $\mathbf{s}$

The quantum subset-sum solver

What we mean by "quantum" here:

A **quantum algorithm** *QSS* which maps ($\mathbf{k}$ being fixed):

$$|v\rangle |\mathbf{b}\rangle \mapsto |v\rangle |\mathbf{b} \oplus QSS(v)\rangle$$

Failures in QSS and cases with more than one solution yield some manageable errors.

Implementing QSS

Solving a subset-sum instance **in superposition**, with poly($\mathbf{n}$) qubits:

- Basic: Grover search in time $\widetilde{\mathcal{O}}\left(2^{\mathbf{n}/2}\right)$
- Using quantum-accessible memory: reuse the procedure of **[BBSS20]**, time $\widetilde{\mathcal{O}}\left(2^{0.2356\mathbf{n}}\right)$
- Using classical memory: reuse the algorithm of **[HM20]**: time $\widetilde{\mathcal{O}}\left(2^{0.428\mathbf{n}}\right)$ and classical space $\widetilde{\mathcal{O}}\left(2^{0.285\mathbf{n}}\right)$

$\Longrightarrow$ improvement: new algorithm in quantum time $\widetilde{\mathcal{O}}\left(2^{0.4165\mathbf{n}}\right)$ and classical space $\widetilde{\mathcal{O}}\left(2^{0.2334\mathbf{n}}\right)$

Bonnetain, Bricout, S., Shen, "Improved classical and quantum algorithms for subset-sum", ASIACRYPT 2020

Helm, May, "The power of few qubits and collisions - subset sum below Grover's bound", PQCrypto 2020

Trade-off

Trade-off by pre-processing

If the labels k_i could have a specific form, the **subset-sum problem could become easier**:

$$
\begin{array}{c}
 \\ k_1 \\ k_2 \\ \vdots \\ k_{m-t} \\ k_{m-t+1} \\ \vdots \\ k_m
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} 1 & 2 & \cdots & m-t & m-t+1 & \cdots & \mathbf{n} \end{array} \\
\begin{pmatrix}
1 & \bullet & \cdots & \bullet & \bullet & \cdots & \bullet \\
0 & 1 & \ddots & \bullet & \bullet & \cdots & \bullet \\
\vdots & \vdots & \ddots & \ddots & \vdots & \cdots & \bullet \\
0 & 0 & \cdots & 1 & \bullet & \cdots & \bullet \\
0 & 0 & \cdots & 0 & \bullet & \cdots & \bullet \\
\vdots & \vdots & & \vdots & \vdots & \cdots & \bullet \\
0 & 0 & \cdots & 0 & \bullet & \cdots & \bullet
\end{pmatrix}
\end{array}
$$

Corner case: no preprocessing

$$\begin{array}{c} \\ k_1 \\ k_2 \\ \vdots \\ k_m \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2\cdots & \mathbf{n} \end{array} \\ \begin{pmatrix} \bullet & \cdots & \bullet \\ \bullet & \cdots & \bullet \\ \vdots & \ddots & \vdots \\ \bullet & \cdots & \bullet \end{pmatrix} \end{array}$$

- No constraints on the labels, the subset-sum problem is hard
- This is our linear-queries algorithm

Corner case: complete preprocessing

$$\begin{array}{c} \\ k_1 \\ k_2 \\ \vdots \\ k_{\mathbf{n}} \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & \mathbf{n} \end{array} \\ \begin{pmatrix} 1 & \bullet & \cdots & \bullet \\ 0 & 1 & \ddots & \bullet \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \end{array}$$

- $(i-1)$-bit constraint on each label k_i, the subset-sum problem is trivial
- This corresponds to Kuperberg 2011: this sequence of labels is constructed in time

$$\sum_i 2^{\sqrt{2i}} = \mathcal{O}\left(\sqrt{\mathbf{n}}2^{\sqrt{2\mathbf{n}}}\right)$$

Conclusion

Conclusion

- New linear-queries algorithm for DCP
- New natural trade-off for sieving algorithms for DCP
- Improvement on the Helm-May algorithm to run Subset-sum in superposition

Unfortunately, quantum subset-sum algorithms are not that fast.
$\implies$ need to improve them to make the linear-queries algorithm competitive for small CSIDH instances.

Thank you!