Time and Query Complexity Tradeoffs for the Dihedral Coset Problem

Maxime Remaud¹, André Schrottenloher², Jean-Pierre Tillich³

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



Introduction •0000 Linear-queries Algorithm

Trade-off

Conclusion

Abelian hidden shift & DCP

Abelian hidden shift

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

 \implies underlies the security of some post-quantum schemes (e.g. CSIDH) $G=\mathbb{Z}_N$ in this talk.

Introduction	Linear-queries Algorithm	Trade-off	Cor
0000	000000	0000	00

Abelian hidden shift & DCP (ctd.)

Quantum 101:

- \bullet operate on states $\sum_{\mathbf{x}} \alpha_{\mathbf{x}} \left| \mathbf{x} \right\rangle$ with complex amplitudes $\alpha_{\mathbf{x}}$
- α_x is not observable, measuring returns x with probability $|\alpha_x|^2$
- Create a superposition:

$$rac{1}{\sqrt{2}}(\ket{0}+\ket{1})rac{1}{\sqrt{\mathsf{N}}}\sum_{x\in\mathbb{Z}_{\mathsf{N}}}\ket{x}$$

$$\frac{1}{\sqrt{2\mathsf{N}}} \ket{0} \sum_{x} \ket{x} \ket{f(x)} + \frac{1}{\sqrt{2\mathsf{N}}} \ket{1} \sum_{x} \ket{x} \ket{g(x)}$$

• Measure the output: get $rac{1}{\sqrt{2}}\left(\ket{0}\ket{x}+\ket{1}\ket{x+s}
ight)$

DCP: Find **s** from such states (with random *x*).

nclusion

Introduction	Linear-queries Algorithm	Trade-off 0000	Conclusion
Ы			

Phase vectors

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

$$\begin{array}{l} \displaystyle \frac{1}{\sqrt{2}} (|x,0\rangle + |x+\mathbf{s},1\rangle) \\ \\ \displaystyle \xrightarrow{QFT_{\mathbf{N}}} \frac{1}{\sqrt{2\mathbf{N}}} \sum_{k \in \mathbb{Z}_{\mathbf{N}}} \omega_{\mathbf{N}}^{kx} \left| k \right\rangle \left(\left| 0 \right\rangle + \omega_{\mathbf{N}}^{ks} \left| 1 \right\rangle \right) \\ \\ \\ \displaystyle \xrightarrow{\text{Measure 1st register}} \frac{1}{\sqrt{2}} \left| k \right\rangle \left(\left| 0 \right\rangle + \omega_{\mathbf{N}}^{ks} \left| 1 \right\rangle \right) =: \left| \psi_k \right\rangle \end{array}$$

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

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

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	000000	0000	00
This naner			

- 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.

⇒ improving the attacks on CSIDH would require **better quantum** algorithms for subset-sum

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	●00000	0000	00

Linear-queries Algorithm

Introductio	n
00000	

Regev's combination routine

Idea:

- combine many $|\psi_{k_i}
 angle$ ($\mathbf{k}=(k_1,\ldots,k_m)$)
- create a new $|\psi_k
 angle$ 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 **b** such that $\mathbf{b} \cdot \mathbf{k} = z$
- 4: If there are two vectors, the state is:

$$\frac{1}{\sqrt{2}} \left(\omega_{\mathsf{N}}^{\mathsf{sb_1} \cdot \mathsf{k}} \left| \mathbf{b}_1 \right\rangle + \omega_{\mathsf{N}}^{\mathsf{sb_2} \cdot \mathsf{k}} \left| \mathbf{b}_2 \right\rangle \right) \underbrace{\simeq}_{\mathsf{global phase}} \frac{1}{\sqrt{2}} \Big(\left| \mathbf{b}_1 \right\rangle + \omega_{\mathsf{N}}^{\mathsf{s}} \underbrace{(\mathbf{b}_2 - \mathbf{b}_1) \cdot \mathsf{k}}_{\mathsf{N}} \left| \mathbf{b}_2 \right\rangle \Big)$$

New Jahel k

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	00000	0000	00
Conseque	nces		

The difficult step: computing b₁ and b₂
 (random) subset-sum problem

given $\mathbf{k} = (k_1, \dots, k_m)$, 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 $m \simeq \log_2 B \simeq \sqrt{\mathbf{n}}$ $\implies \simeq \sqrt{\mathbf{n}}^{\sqrt{\mathbf{n}}}$ operations & queries
- We can also combine \simeq **n** phase vectors to recover one bit of **s**, and repeat

$\Rightarrow \simeq n^2$ 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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	000000	0000	00
New algo	orithm		

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 $\boldsymbol{k}\cdot\boldsymbol{b}\mod\boldsymbol{N}$

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

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

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

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	000000	0000	00

The quantum subset-sum solver

What we mean by "quantum" here:

A quantum algorithm QSS which maps (k being fixed):

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

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	○○○○○●	0000	
Implementing	QSS		

Solving a subset-sum instance in superposition, with poly(n) qubits:

- \bullet Basic: Grover search in time $\widetilde{\mathcal{O}}\left(2^{n/2}\right)$
- Using quantum-accessible memory: reuse the procedure of [BBSS20], time $\widetilde{\mathcal{O}}\left(2^{0.2356n}\right)$
- Using classical memory: reuse the algorithm of [HM20]: time $\widetilde{\mathcal{O}}(2^{0.428n})$ and classical space $\widetilde{\mathcal{O}}(2^{0.285n})$
- $\implies \text{ improvement: new algorithm in quantum time } \widetilde{\mathcal{O}}\left(2^{0.4165n}\right) \text{ and } \\ \text{ classical space } \widetilde{\mathcal{O}}\left(2^{0.2334n}\right)$

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	000000	●000	00

Trade-off

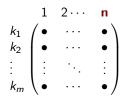
Introduction 00000	Linear-queries Algorithm	Trade-off O●OO	Conclusion
Trade-off by	pre-processing		

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

	1	2	• • •	m-t	m-t+1		n
k_1	(1)	٠		•	٠		• \
k ₂	0	1	·	•	•		•
÷	:	÷	·	·	•		•
k_{m-t}	0	0	•••	1	•		•
k_{m-t} k_{m-t+1}	0	0	• • •	0	•	• • •	•
:	1	÷		÷	:		•
k _m	0/	0	•••	0	•		•/

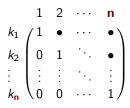
00000	000000	0000	00
Introduction		Trade-off	Conclusion

Corner case: no preprocessing



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

Introduction 00000	Linear-queries Algorithm	Trade-off 000●	Conclusion
Corner case:	complete prep	rocessing	



- (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

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

Introduction	Linear-queries Algorithm	Trade-off	Conclusion
00000	000000	0000	• 0

Conclusion

Introduction	Linear-queries Algorithm	Trade-off 0000	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!