Some remarks on CSIDH and CSI-FiSh

Steven Galbraith

University of Auckland, New Zealand



(Rangitoto Island, a volcano)

Thanks

Thanks to Fre and Wouter.

December 2024, Auckland, New Zealand

The AMS-AustMS-NZMS joint meeting will take place December 9-13, 2024 in Auckland.

https://ms-meet-2024.blogs.auckland.ac.nz/

There will be a Special Session on Computational Number Theory and Applications
(Organisers: David Harvey, Drew Sutherland, and me)

Plan

- Relation lattices and CSIDH.
- Quantum Reduction Vectorization to Parallelization
- Montgomery and Zhandry (Asiacrypt 2022)
- New method (joint work with Yi-Fu Lai and Hart Montgomery)

Please interrupt and ask questions at any time.

Relation lattices

- Let G be a finite abelian group, e.g., $\mathbb{Z}_p^*$.
- Let $g_1, \ldots, g_n \in G$.
- The relation lattice is

$$L = \left\{ (x_1, \ldots, x_n) \in \mathbb{Z}^n : \prod_{i=1}^{n} g_i^{x_i} = 1 \right\}.$$

- CSIDH/CSI-FiSh relation lattice: G is the ideal class group of $\mathbb{Z}[\sqrt{-p}]$ and g_i are split prime ideals $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ of small distinct prime norms.
- Finding close vectors in relation lattice important for many applications of CSIDH/SCALLOP/etc
(not key exchange/encryption, but signatures etc).

Every lattice is a relation lattice

- Let $L \subseteq \mathbb{Z}^n$ have rank n.
- Then $G = \mathbb{Z}^n / L$ is a finite abelian group, and L is a relation lattice for G.
- My interest is relation lattices where the group G has interesting mathematical structure (e.g., a notion of *smoothness*) and where the chosen elements $g_1, \ldots, g_n \in G$ are *special* in some way.
  Examples: Ducas and Pierrot (2019), Li, Ling, Xing and Yeo (2020).

Features of CSIDH relation lattice (1)

- Let $\underline{x} = (x_1, \ldots, x_n) \in L$ be non-zero. So

$$\prod_{i=1}^{n} \mathfrak{l}_i^{x_i} = 1.$$

- Consider the ideal

$$\left(\prod_{i, x_i > 0} \mathfrak{l}_i^{x_i} \right) \left(\prod_{i, x_i < 0} \bar{\mathfrak{l}}_i^{|x_i|} \right) = (A, B + \sqrt{-p})$$

  where $A = \prod_i \ell_i^{|x_i|}$.
- This is a principal ideal, so $(A, B + \sqrt{-p})$ not reduced.
- Hence $A > \sqrt{4p/3}$ by reduction theory.
- Approximating $\ell_i = n \ln(n)$ gives $\|\underline{x}\|_1 > \frac{1}{2} \log(p) / \log(n \ln(n))$.
- So we have a lower bound on the shortest non-zero vector.

Features of CSIDH relation lattice (2)

- Bounded Distance Decoding (BDD): Let $\underline{v}$ have distance at most $\lambda_1(L)/2$ from a lattice point. Compute the closest lattice point.
- Let $\underline{v} = \underline{x} + \underline{e}$ where $\underline{x} \in L$ and $\|\underline{e}\|_1 < \frac{1}{2}\log(p)/\log(n\ln(n))$.
- Then

$$\prod_{i=1}^{n} \mathfrak{l}_i^{v_i} = \prod_{i=1}^{n} \mathfrak{l}_i^{e_i}.$$

- Now, as on previous slide, this ideal is equivalent to

$$\left(\prod_{i,e_i>0} \mathfrak{l}_i^{e_i}\right)\left(\prod_{i,e_i<0} \bar{\mathfrak{l}}_i^{|e_i|}\right) = (A, B+\sqrt{-p})$$

  where $A = \prod_i \ell_i^{|e_i|}$.
- Then $(A, B+\sqrt{p})$ is a reduced ideal, and factoring gives $|e_i|$.
- So we have a poly-time BDD algorithm.

Solving CVP in the CSIDH relation lattice

- There exists a polynomial-time BDD algorithm for the CSIDH lattice, for vectors that are very close in the ℓ_1-norm to a lattice point.
- But no efficient CVP algorithm is known for the CSIDH lattice.

Solving CVP in the CSIDH relation lattice

- **Open question:** Are there more efficient algorithms for CVP for the CSIDH lattice?
- If not, then asymptotically the best signature scheme is SeaSign.
- Note: Efficient smoothing of ideal classes without knowing relation lattice first would lead to efficient algorithm to compute class numbers and solve DLP in class groups.

Group action problems

- G a group acting on set X.
- For simplicity assume G cyclic and $|X| = |G|$ and action transitive.
- Vectorization: Given $(x, g * x)$ to compute g.
- Parallelization: Given $(x, a * x, b * x)$ to compute $(ab) * x$.

Quantum Reduction Vectorization to Parallelization (G, Panny, Smith, Vercauteren, 2021)

- Let $\mathcal{A}$ be an circuit such that $\mathcal{A}(a * x, b * x) = (ab) * x$
- Assumes:
  (1) effective group action (EGA);
  (2) circuit $\mathcal{A}$ is always correct.
- Given $(x, a * x)$ can use circuit as
  $\mathcal{A}(a^m * x, a^n * x) = (a^{m+n}) * x$.
- Let g generate cyclic group G.
- Define $f : \mathbb{Z}^2 \to X$ by $f(s, t) = (g^s) * (a^t) * x$.
- Shor/Hidden subgroup returns element of lattice
  $L = \{(s, t) \in \mathbb{Z}^2 : g^s a^t = 1\}$.
- If $\gcd(t, |G|) = 1$ then solve DLP of a and hence can compute a.
- Note: Need a quantum circuit to compute $f(s, t)$.

Quantum Reduction Vectorization to Parallelization (G, Panny, Smith, Vercauteren, 2021)

- Project from a brainstorming session in 2018 in Leuven.
- "While the paper's main result is worth to be published, it is questionable whether the ideas here are brilliant enough to warrant a 5-page crypto paper."
- "The problem studied in this paper is interesting. However, the major downside is that the reduction only works when given a perfect CDHP solver, which is too restricted to have significant crypto implications."
- "The paper can be critiqued in several respects: first of all, this paper literally raises the question what a least publishable unit is. Whilst to me this is mathematically correct, cunning observation, it is not enough for a paper at a high profile conference."

Montgomery-Zhandry (2022)

- Handle oracle that is only correct with probability ϵ.
- Best paper at Asiacrypt 2022.
- For $y, z \in X$ define

$$\mathcal{A}_0(y, z) = (a'b')^{-1} * \mathcal{A}(a' * y, b' * z).$$

- Let $\mathcal{D}$ be output distribution of $\mathcal{A}_0(x, x)$.
- Lemma:
  (1) $\Pr(x \leftarrow \mathcal{D}) = \epsilon$;
  (2) output distribution of $\mathcal{A}_0(a * x, b * x)$ is $(ab) * \mathcal{D}$.

Example

- Let $g \in G$ have order 3.
- Suppose $\mathcal{A}(a * x, b * x)$ returns $(ab) * x$ with probability $1/3$, $(gab) * x$ with probability $1/3$, and x with probability $1/3$.
- Then $\mathcal{A}_0(a * x, b * x)$ returns $(ab) * x$ with probability $1/3$, $(gab) * x$ with probability $1/3$, and uniformly distributed elements $(a'b')^{-1} * x$ with probability $1/3$.
- Let L be the list of outputs and $\tilde{D}$ the resulting distribution. So there will be two set elements w_1, w_2 that appear with approx $1/3$ frequency.
- Exercise: If $w = (gab) * x$ and we call $\mathcal{A}_0(x, w)$ for T times to get $\tilde{D}_w$ then what is $\Delta(\tilde{D}, \tilde{D}_w)$ for your favourite statistical distance function Δ?

Montgomery-Zhandry - Algorithm $\mathcal{A}_1(y, z)$

- Run $\mathcal{A}_0(y, z)$ for T times to get a list S of outputs (some may be multiple times) and an estimate $\tilde{\mathcal{D}}$ of distribution $\mathcal{D}$.
- When $T \gg 1/\epsilon$ then $(ab) * x$ will be on the list.
- Then for each $w \in S$ run $\mathcal{A}_0(x, w)$ for T times.
- When $w = (ab) * x$ then the resulting distribution will be the same as $\mathcal{D}$.
- Define $\mathcal{D}_w$ to be the output distribution of $\mathcal{A}_0(x, w)$, and $\tilde{\mathcal{D}}_w$ our estimate of it.
- Let $\Delta(\tilde{\mathcal{D}_w}, \tilde{\mathcal{D}})$ be a statistical distance.
- Store in a list L all w for which $\Delta(\tilde{\mathcal{D}_w}, \tilde{\mathcal{D}})$ is small.
- Return L

Montgomery-Zhandry - Subgroups

- A major insight of Montgomery-Zhandry is that there can be a subgroup H of G of order at most $1 + 1/\epsilon$ which is an obstruction to this approach.
- It means L contains not just $(ab) * x$ but also $(hab) * x$ for some/all $h \in H$.

Montgomery-Zhandry full algorithm

- Input $(x, a * x)$ and quantum access to circuit $\mathcal{A}$.
- Determine H.
- Algorithm $\mathcal{A}_1(a * x, b * x)$ outputs a list L that contains $(ab) * x$ and maybe more.
- Algorithm $\mathcal{A}_2(a * x, b * x)$ "fills out" subgroup so that it outputs set $\{(hab) * x : h \in H\}$.
- Run GPSV with respect to action of G/H on X/H. Here G/H and X/H are represented in $O(1 + 1/\epsilon)$ space as cosets/orbits.
- Once have computed coset aH do brute force over all $g \in H$ to determine a.
- Complexity: Paper claims number of queries to $\mathcal{A}$ at least $1/\epsilon^{13}$, but actually at least $1/\epsilon^{21}$.

New approach (with Yi-Fu Lai and Hart Montgomery)

- Let $\mathcal{D}$ be output distribution of $\mathcal{A}_0(x, x)$.
- Write the support of $\mathcal{D}$ as $x_1, x_2, \ldots$ with $p_i = Pr(x_i \leftarrow \mathcal{D})$ such that $p_1 \geq p_2 \geq p_3 \geq \cdots$.
- We prove a "thresholding lemma": there is an integer $i < O(1/\epsilon)$ such that $p_i \leq \epsilon$ and $p_i - p_{i+1} > O(1/\epsilon^2)$.
- Let $S = \{x_1, \ldots, x_i\}$ be these "heavy" elements.
- There exists an algorithm $\mathcal{A}_1(y, z)$ that with overwhelming probability computes exactly this heavy set.
- With overwhelming probability contains $(ab) * x \in S$.
- Running time of $\mathcal{A}_1$ is $O(\lambda/\epsilon^3)$ queries to $\mathcal{A}$.

New approach

- Algorithm $\mathcal{A}_1(y, z)$ does the thresholding and outputs a set $\mathcal{S}$ of size $O(1/\epsilon)$.
- $(ab) * x \in \mathcal{S}$ with overwhelming probability.
- $\mathcal{A}_2(\mathcal{S})$ prunes $\mathcal{S}$:
  For each $w \in \mathcal{S}$ run $\mathcal{A}_1(x, w)$ and if the output is equal to $\mathcal{S}$ then store w in a list L.
- Then L contains $(ab) * x$.
- Can show that $L = \{(hab) * x : h \in H\}$ for some subgroup H.
- When G is cyclic the subgroup H can be classically pre-computed using $\mathcal{A}$.
- For general G, need quantum hidden subgroup algorithm to compute H.
- Total cost of reduction is $O(\lambda/\epsilon^4)$ queries to $\mathcal{A}$.

Example

- Consider our earlier example.
- $g \in G$ has order 3 and $\mathcal{A}_0(a * x, b * x)$ returns $(ab) * x$ with probability $1/3$, $(gab) * x$ with probability $1/3$, and uniformly distributed elements with probability $1/3$.
- There is a big gap between elements that appear with probability $1/3$ and elements that appear with probability $1/|G|$. So the thresholding algorithm reliably computes

$$S = \{(ab) * x, (gab) * x\}.$$

- When run $\mathcal{A}_1(x, (ab) * x)$ get the same set S.
- When run $\mathcal{A}_1(x, (gab) * x)$ then we get the set $\{(gab) * x, (g^2ab) * x\} \neq S$.
- Hence $L = \{(ab) * x\}$ is the output of $\mathcal{A}_2$.

Example: Hard case - discrete Gaussians

- Suppose G cyclic with prime order. Fix $g \in G$.
- Let $\mathcal{D}$ be a discrete Gaussian on $\mathbb{Z}$ with parameter c/ϵ for suitable constant c so that $\Pr(0 \leftarrow \mathcal{D}) = \epsilon$.
- Suppose oracle outputs $(g^i ab) * x$ where i is sampled from $\mathcal{D}$.
- There are $O(1/\epsilon)$ "heavy" elements with frequency close to $1/\epsilon$.
- Observation: $\Delta(\mathcal{D}, i + \mathcal{D}) = \Omega(i^2\epsilon^2)$ for small $i \in \mathbb{Z}$, where Δ is the squared-Hellinger distance.
- It is a theorem in statistical testing that cannot distinguish with fewer than $\Omega(1/\Delta(\mathcal{D}, i + \mathcal{D}))$ samples when Δ is the squared-Hellinger distance.
- Hence need at least $\Omega(1/(i^2\epsilon^2))$ queries to $\mathcal{A}$ to distinguish $\mathcal{D}$ from $i + \mathcal{D}$.

On REGAs

- Due to the lattice problems, CSIDH is a restricted effective group action (REGA).
- Montgomery-Zhandry introduce the 1D-inhomogeneous SIS (1D-ISIS) problem for group actions.
- They show that if one can solve this, then one has an EGA from an REGA.
- They also prove that if there is an efficient adversary that can solve the group action DLog problem on regular REGAs then there is an efficient algorithm to solve the 1D-SIS problem. (Lemma 35 of their paper.)

Summary and open problems

Summary:

- Montgomery-Zhandry give an algorithm to turn a parallelisation circuit $\mathcal{A}$ that is correct with probability $1/\epsilon$ into an overwhelmingly correct parallelisation circuit that makes $O(\lambda/\epsilon^{21})$ queries to $\mathcal{A}$.
- We improve this to $O(\lambda/\epsilon^4)$, with a much simpler algorithm and proofs.
- We give evidence that $O(\lambda/\epsilon^2)$ queries is required.

Open problems:

- Get an efficient group action (EGA) for CSIDH.
- Improve statistical testing tools and hence get a better reduction.

Thank You





