Finding isogenies of fixed degree between supersingular elliptic curves

Benjamin Benčina, Péter Kutas, Simon-Philipp Merz, Christophe Petit,
Miha Stopar, Charlotte Weitkämper

December 12th, 2023

Isogeny Club - Season Three

Contents

Isogeny problems

When looking at supersingular elliptic curves, naturally questions about isogenies between specific curves arise:

- The *pure isogeny problem*: find any isogeny between given supersingular elliptic curves.
- The *SIDH variant*: find an isogeny of specific degree and torsion action between given supersingular curves.
- The *fixed-degree variant*: find an isogeny of specific degree between given supersingular curves.

The problem of finding fixed-degree isogenies

Problem

Let p be a prime, and E_1 and E_2 supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Let d be a positive integer. Find an isogeny $E_1 \to E_2$ of degree d.

We want to examine the general problem where (other than size) there are no restrictions on the degree d.

Let $\epsilon > 0$ be such that $d \approx p^{1/2+\epsilon}$.

The state of the art

Computing endomorphism rings takes

- $O^*(p^{1/2})$ classically.
- $O^*(p^{1/4})$ quantumly with Grover.

Computing fixed-degree isogenies classically via

- *exhaustive search* over all outgoing isogenies: cost $O(d)$.
- *meet-in-the-middle*: cost $O^*(\sqrt{d})$ time and memory.
- *van Oorschot–Wiener collision search variants*: cost depends heavily on available memory.

Quantum speed-ups:

- Grover's algorithm improves exhaustive search to $O^*(\sqrt{d})$.
- (Tani's algorithm: $d^{1/3}$.)

The state of the art

Computing endomorphism rings takes

- $O^*(p^{1/2})$ classically.
- $O^*(p^{1/4})$ quantumly with Grover.

Computing fixed-degree isogenies classically via

- *exhaustive search* over all outgoing isogenies: cost $O(d)$.
- *meet-in-the-middle*: cost $O^*(\sqrt{d})$ time and memory.
- *van Oorschot–Wiener collision search variants*: cost depends heavily on available memory.

Quantum speed-ups:

- Grover's algorithm improves exhaustive search to $O^*(\sqrt{d})$.
- (Tani's algorithm: $d^{1/3}$.)

The state of the art

Computing endomorphism rings takes

- $O^*(p^{1/2})$ classically.
- $O^*(p^{1/4})$ quantumly with Grover.

Computing fixed-degree isogenies classically via

- *exhaustive search* over all outgoing isogenies: cost $O(d)$. *(general d, but specifically primes)*
- *meet-in-the-middle*: cost $O^*(\sqrt{d})$ time and memory. *(smooth d)*
- *van Oorschot–Wiener collision search variants*: cost depends heavily on available memory. *(smooth d)*

Quantum speed-ups:

- Grover's algorithm improves exhaustive search to $O^*(\sqrt{d})$. *(general d, but specifically primes)*
- (Tani's algorithm: $d^{1/3}$.)

Our strategy

1. Compute the endomorphism rings of E_1 and E_2.
2. Construct a connecting ideal between these two quaternion orders.
3. Compute the norm form associated to $Hom(E_1, E_2)$.
4. Represent d via this norm form.
5. Compute an ideal equivalent to the connecting ideal of correct norm.
6. Convert the ideal back to an isogeny representation.

Individual steps I

1. Compute the endomorphism rings of E_1 and E_2.
   $\implies$ O_1 and O_2 can be found using Eisenträger et al. (and Grover for quantum speed up).
2. Construct a connecting ideal between O_1 and O_2.
   $\implies$ Kirschmer and Voight provide an efficient algorithm for finding I.
3. Compute the norm form associated to $Hom(E_1, E_2)$.
   $\implies$ First compute an LLL-reduced Gram matrix G of the ideal I, where $g_{ij} = \langle \sigma_i, \sigma_j \rangle = tr(\sigma_i \overline{\sigma_j})$ for σ_i a basis of I. Then normalise the matrix by $Norm(I)$ and compute the associated norm form Q:

$$Q(x_1, x_2, x_3, x_4) = (x_1\ x_2\ x_3\ x_4)\, G\, (x_1\ x_2\ x_3\ x_4)^T$$

4. Represent d via this norm form.
   $\implies$ Find a solution to $Q(x_1, x_2, x_3, x_4) = d$, where Q is a quadratic form and we have bounds on the x_i.

Individual steps II

5. Compute an ideal equivalent to the connecting ideal of correct norm.
 $\implies$ Like in KLPT can compute J with of norm d such that $J \approx I$.
6. Convert the ideal back to an isogeny representation.
 $\implies$ Depending on d, this can mean a sequence of rational maps or a representation like Robert's for non-smooth degrees.

Our main task

1. Compute the endomorphism rings of E_1 and E_2.
2. Construct a connecting ideal between these two quaternion orders.
3. Compute the norm form associated to $Hom(E_1, E_2)$.
4. Represent d via this norm form.
   $\implies$ Find a solution to $Q(x_1, x_2, x_3, x_4) = d$, where Q is a quadratic form and we have bounds on the x_i.
5. Compute an ideal equivalent to the connecting ideal of correct norm.
6. Convert the ideal back to an isogeny representation.

Solve Step 4 given that $Norm(\sigma_i) \approx \sqrt{p}$ for σ_i with $i = 1, \ldots, 4$ an LLL-reduced basis of I and $|x_i| < c \cdot p^{\epsilon/2}$.

Solving Step 4: Cornacchia's algorithm I

- Need to find a solution to multivariate equation $Q(x_1, x_2, x_3, x_4) = d$.
- From the way a basis of the lattice I is computed, we have bounds on $|x_i|$.
- Guess two variables, say $k = x_3$ and $l = x_4$.
- Thus we want to solve the equation

$$\begin{aligned} f(x_1, x_2) &= Q(x_1, x_2, k, l) - d \\ &= g_{11}x_1^2 + g_{22}x_2^2 + 2g_{12}x_1x_2 && \text{(quadratic)} \\ &+ (2g_{13}k + 2g_{14}l)x_1 + (2g_{23}k + 2g_{24}l)x_2 && \text{(linear)} \\ &+ \left(2g_{34}kl + g_{33}k^2 + g_{44}l^2 - d\right), && \text{(constant)} \end{aligned}$$

Solving Step 4: Cornacchia's algorithm II

- Changing variables transforms

$$f(x_1, x_2) = Q(x_1, x_2, k, l) - d$$

  into an equation of the form

$$x^2 - Dy^2 = N$$

  which can be solved with Cornacchia's algorithm given that N does not have too many prime factors as we need to factor N to find all square roots of $D(mod N)$.
- If we do not find a solution we make another guess for (x_3, x_4).
- We can show that if N has at most $B \log\log N$ distinct prime divisors for $B = 11$, we obtain a solution in $> 99\%$ of cases after working through all guesses. Abandoning N with more prime divisors leads to a small failure probability.
- **Complexity**: quantum time $O^*(p^{\epsilon/2})$, $O^*(p^{\epsilon}) \cdot L_{\log p}(1/3)$ classically, or the algorithm returns no solution.

Solving Step 4: Coppersmith's algorithms I

- Again, need to find a solution to multivariate equation $Q(x_1, x_2, x_3, x_4) = d$, and guess one or two variables.
- Using Coppersmith variants due to Coron and Bauer–Joux, we want to solve the remaining bivariate or trivariate equation.
- Restrictions on the size of ϵ arise from Coppersmith limitations.
- Bivariate Coron complexity: $O^*(p^{\epsilon})$ classically or $O^*(p^{\epsilon/2})$ on a quantum computer when $\epsilon < 1/2$.
- Trivariate Bauer–Joux complexity: $O^*(p^{\epsilon/2})$ classically or $O^*(p^{\epsilon/4})$ on a quantum computer.

More guessing: a hybrid approach

- If the degree d is sufficiently smooth, we can additionally guess parts of the isogeny starting from E_1 or E_2 to decrease the parameter sizes of the norm equation.
- Let $d = \ell^e \approx p^{1/2+\epsilon}$ such that ϵ is too large for the other methods to work efficiently.
- New strategy:
  1. Guess ℓ^{e_1}-isogeny $\phi_1 : E_1 \to E$.
  2. Use ϕ_1 to compute $End(E)$ from $End(E_1)$.
  3. Solve the fixed-degree isogeny problem with E and E_2 for degree ℓ^{e-e_1} to obtain ϕ_2, or guess again.
  4. Compose ϕ_2 with ϕ_1 to find a solution to the original problem.
- Classically we obtain a complexity of $O^*(\max\{p^{1/2}, p^{\epsilon-1/8}\})$ with Coppersmith's trivariate method.

Cost of our algorithms (in $\log_p$)

Method	Cost (classical)	Cost (quantum)	Condition on size
State of the art (general d)	$\frac{1}{2}+\epsilon$	$\frac{1}{4}+\frac{\epsilon}{2}$	
Cornacchia (our version)	$\max\{\frac{1}{2}, 2\epsilon\}$	$\max\{\frac{1}{4}, \epsilon\}$	
Coppersmith bivariate	$\max\{\frac{1}{2}, 2\epsilon\}$	$\max\{\frac{1}{4}, \epsilon\}$	$\epsilon < 1/4$
Coppersmith trivariate	$\max\{\frac{1}{2}, \epsilon\}$	$\max\{\frac{1}{4}, \frac{\epsilon}{2}\}$	$\epsilon < 0.16$
Hybrid approach (smooth d)	$\max\{\frac{1}{2}, \epsilon-\frac{1}{8}\}$	$\max\{\frac{1}{4}, \frac{\epsilon}{2}\}$	$\epsilon > 1/4$

Results

Smooth degrees (classical)

- Comparison to MITM with $p^{1/4+\epsilon/2}$.
- We always compute endomorphism rings, so we consider $\epsilon > 1/2$.
- The hybrid algorithm works best in ranges $p \leq d \leq p^{5/4}$.
- MITM has large memory-requirements, while our algorithms are low-memory and parallelisable.

Non-smooth degrees (classical)

- All methods have same complexity.
- For ranges $\sqrt{p} < d < p^3$, any algorithm improves upon the state of the art.

Quantum algorithms

- No difference between smooth and non-smooth.
- For ranges $\sqrt{p} < d < p^3$, the Cornacchia approach is fastest.
- For ranges $\sqrt{p} < d < p$, bivariate Coppersmith is preferable (no heuristics).

Summary

- We provide improved algorithms for computing d-isogenies using Cornacchia's algorithm and Coppersmith methods to solve Diophantine equations.
- Further improvements can stem from hybrid algorithms utilising Coppersmith's trivariate algorithm.
- The Cornacchia approach has no condition on the size of the parameters but requires a small heuristic.
- The Coppersmith approaches have conditions on the size of the degree but require no heuristics.
- We improve isogeny finding where $d = p^{1/2+\epsilon}$ for $1/2 < \epsilon < 5/2$ in different settings.

Open questions & further ideas

- Perform more experiments.
- Can these algorithms be utilised constructively?
- Work on Coppersmith variants which do not involve any guessing (solve the four-variable equation directly).

Thank you!

References I

- Bauer, Aurélie and Joux Antoine. "Toward a Rigorous Variation of Coppersmith's Algorithm on Three Variables". In: *Advances in Cryptology — EUROCRYPT 2007*. 2007.
- Coppersmith, Don. "Finding a small root of a bivariate integer equation; factoring with high bits known". In: *Advances in Cryptology — EUROCRYPT 1996*. 1996.
- — ."Finding a small root of a univariate modular equation". In: *Advances in Cryptology — EUROCRYPT 1996*. 1996.
- Cornacchia, Giuseppe. "Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$". In: *Giornale di Matematiche di Battaglini* (1908).
- Coron, Jean-Sébastien. "Finding small roots of bivariate integer polynomial equations: A direct approach". In: *Advances in Cryptology — CRYPTO 2007*. 2007.
- Eisenträger, Kirsten et al. "Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs". In: *Open Book Series* (2020).

References II

Galbraith, Steven D. et al. "On the Security of Supersingular Isogeny Cryptosystems". In: *Advances in Cryptology - ASIACRYPT 2016*.

Grover, Lov K. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996.

Kohel, David et al. "On the quaternion ℓ-isogeny path problem". In: *LMS Journal of Computation and Mathematics* (2014).