Munich Isogeny Seminar, Remote, 18th of December 2025

A presentation on work in progress

Restricted Isogeny Class-Group Action Computation in Dimension 2

Joint with Andrea Basso, Giacomo Borin and Sina Schaeffler

Ryan Rueger
IBM Research Zurich & Technical University of Munich

In one slide

We present an algorithm to sample elements from the class-group such that their action can be computed practically efficiently in dimension 2 (CSIDH does this in dimension-1)

This sampling algorithm samples from a heuristically exponentially-sized subset of the class group (and appears to be randomly distributed)

However the resulting action is restricted (since we can only compute in dim-2 for our specially sampled ideals)

Although the sampling algorithm is slow (work in progress!)
the resulting action computation is competitive with CSIDH (slow “keygen” is okay for NIKE)

and it is compatible with unrestricted PEGASIS (we can combine fast(er) restricted action with slow(er) unrestricted actions for more complex protocols)

Unrestricted Computation: Clapoti

Let E be a supersingular Elliptic Curve defined over $\mathbb{F}_p$ (*i.e.* $\mathbb{Z}[\pi]$-oriented)

If the $\mathbb{Z}[\pi]$-ideals $\mathfrak{a}, \overline{\mathfrak{b}}$ represent the same class in $\mathrm{Cl}(\mathbb{Z}[\pi])$, then the commutative diagram

$$\begin{array}{ccc} E & \xrightarrow{\varphi_{\mathfrak{a}}} & E_{\mathfrak{a}} \\ {\scriptstyle \varphi_{\mathfrak{b}}}\downarrow & & \downarrow{\scriptstyle \psi_{\mathfrak{b}}} \\ E_{\mathfrak{b}} & \xrightarrow[\psi_{\mathfrak{a}}]{} & E \end{array} \quad \text{induces the } \textit{Kani-map} \quad \Phi = \begin{pmatrix} \varphi_{\mathfrak{a}} & \widetilde{\psi_{\mathfrak{b}}} \\ -\varphi_{\mathfrak{b}} & \widetilde{\psi_{\mathfrak{a}}} \end{pmatrix} : E \times E \to E_{\mathfrak{a}} \times E_{\mathfrak{b}}$$

of polarised degree $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b})$

If $\mathfrak{a}, \overline{\mathfrak{b}}$ have coprime norms, the kernel of Φ is given by

$$\ker(\Phi) = \{(\mathrm{N}(\mathfrak{a})P, \gamma(P)) \mid P \in E[N]\} \quad \text{where} \quad \gamma\mathbb{Z}[\pi] = \mathfrak{a}\mathfrak{b}$$

Efficiency We require $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$ where $2^e \mid p + 1$

- ...so that the kernel $\ker(\Phi)$ is $\mathbb{F}_{p^2}$-rational ("Usual HD-requirement")

Easy Given $\mathfrak{a}$, finding equivalent $\mathfrak{b}$ with norm coprime is easy

Hard Given $\mathfrak{a}$, finding equivalent $\mathfrak{b}$ with $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) = 2^e$

- qt-PEGASIS solves this by going to dimension 4, allowing for $N = \mathrm{N}(\mathfrak{a}) + \mathrm{N}(\mathfrak{b}) + \mathrm{N}(\mathfrak{c}) + \mathrm{N}(\mathfrak{d})$
- SQISign2D-West uses full endomorphism ring, allowing for $N = u\mathrm{N}(\mathfrak{a}) + v\mathrm{N}(\mathfrak{b})$ (now $\mathfrak{a}, \mathfrak{b}$ quaternionic)

Restricted Computation: CSIDH and QFESTA

What if we don't need unrestricted computation?

CSIDH evaluates ideals that are the product of small ideals

QFESTA evaluates (quaternionic) ideals by "splitting quaternionic endomorphisms"

Slogan "This work evaluates ideals by splitting quadratic endomorphisms"

In Clapoti, one needs equivalent ideals $\mathfrak{a}, \mathfrak{b}$
Such that $\mathrm{N}(\mathfrak{a}) + \mathrm{N}(\overline{\mathfrak{b}}) = 2^e$ and $\gamma\mathcal{O} = \mathfrak{a}\mathfrak{b}$
That is, $\mathrm{N}(\gamma) = \mathrm{N}(\mathfrak{a})(2^e - \mathrm{N}(\mathfrak{a}))$

Converse is also true
Given an endomorphism γ with $\mathrm{N}(\gamma) = q(2^e - q)$ and q odd, we get the $\mathcal{O}$-ideals

$$\mathfrak{a} = (q, \gamma)\mathcal{O}, \overline{\mathfrak{b}} = (2^e - q, \gamma)\mathcal{O} \quad \text{with} \quad \mathrm{N}(\mathfrak{a}) = 2^e, \mathrm{N}(\mathfrak{b}) = 2^e - q$$

Moreover, they are equivalent

$$\mathfrak{a}\mathfrak{b} = (q(2^e - q), q\gamma, (2^e - q)\gamma, \gamma^2)\mathcal{O} = \gamma\mathcal{O}$$

Goal Find element of $\mathcal{O}$ with norm $q(2^e - q)$

The norm equation

Goal Find element γ of $\mathbb{Z}[\pi]$ with norm $q(2^e - q)$

Setting $q = 2^{e-1} - a$, we want

$$\mathrm{N}(\gamma) \overset{!}{=} (2^{e-1} - a)(2^e - (2^{e-1} - a)) = (2^{e-1} - a)(2^{e-1} + a) = 2^{2(e-1)} - a^2$$

Writing $\gamma = x + \pi y$, we have $\mathrm{N}(\gamma) = x^2 + py^2$ and get the norm equation

$$x^2 + a^2 = 2^{2(e-1)} - py^2$$

Suggests The following algorithm

1. Randomly sample an integer $0 \leq y < 2^{e-1}/p^{1/2}$ (so that $0 \leq 2^{2(e-1)} - py^2$)
2. Attempt to factor $L = 2^{2(e-1)} - py^2 = p_1 \cdots p_n$ (*e.g.* by using trial divisions + primality test)
3. Reject if any $p_i \equiv 3 \pmod 4$ appears with odd multiplicity (because then L is not a sum of two squares)
4. Decompose every $p_i = \mu_i^2 + \lambda_i^2$ (using Cornacchia's algorithm)
5. Finally combine solutions to obtain $L = x^2 + a^2$ (using identity $(\mu_1^2 + \lambda_1^2)(\mu_2^2 + \lambda_2^2) = (\mu_1\lambda_2 - \lambda_1\mu_2)(\mu_1\mu_2 + \lambda_1\lambda_2)$)

Now Given a solution x, y, a to the norm equation, we have everything we said we want

$$\gamma = x + \pi y \qquad \mathfrak{a} = (q, \gamma)\mathbb{Z}[\pi] \qquad \overline{\mathfrak{b}} = (2^e - q, \gamma)\mathbb{Z}[\pi] \qquad \mathfrak{a}\mathfrak{b} = (q(2^e - q), q\gamma, (2^e - q)\gamma, \gamma^2)\mathcal{O} = \gamma\mathcal{O}$$

Security: What is the distribution of these ideals?

Recall

We sample $y \leftarrow [0, 2^{e-1}/p^{1/2})$ and try to decompose $L = 2^{2(e-1)} - py^2 = x^2 + a^2$

We return the $\mathbb{Z}[\pi]$-ideal $\mathfrak{a} = (2^{e-1} - a, x + y\pi)$ of norm $q = 2^{e-1} - a$

Strategy for Analysis

Probaility that a class represents both q and $2^e - q$ is $1/h^2$

...but there are $2^{e-1} \sim h^2$ possible q's, giving expected value 1

Size of the set (Assuming sampling is injective $y \xrightarrow{\mathfrak{a}}$

We are sampling y from a set of size $2^{e-1}/p^{1/2}$

The resulting $L = 2^{2(e-1)} - py^2$ is of size $2^{2(e-1)}$ and is 1 mod 4 (when $p \equiv 3 \pmod 4$)

The probability that L is the sum of two squares is $1/\sqrt{2(e-1)}$

So we have $2^{e-1}/p^{1/2}/\sqrt{2(e-1)}$ elements

Idea Pick smaller e: we only need $2^{e-1}/p^{1/2}/\sqrt{2(e-1)} \approx 2^{2\lambda}$

...results in shorter isogeny chains $\log(p) \to \log(p)/2 + 2\lambda$

...and smaller numbers for primality testing $p^2 \to 2^{2\lambda}p$

Experiments

For $\log_2(p) = 31$, $p = c2^f - 1$ and $e = f - 1$ the sampled ideals are all distinct

For $\log_2(p) = 31$, $p = c2^f - 1$ and $e = f + 5$ we cover the entire class group

...and the distribution appears to be fairly uniform

Computing the 2 Dimensional Isogeny

We require E to be on the surface to be able to work over $\mathbb{F}_p$ (more later)

...but then we encounter diagonal isogenies (Like in Qlapoti)

- These cannot be computed in the Theta Model
- ...even x-only is difficult because the kernel points are unfavourable (New work has fixed this)
- This can be seen from the shape of the kernel
- If a 2-isogeny is between a product of elliptic curves it is either diagonal
- ...or is a matrix of automorphisms

The number of diagonal steps is the 2-adic valuation of $((q+x)^2 + py^2)/4$

In practice we see this is very small

Working over $\mathbb{F}_p$

Recall When $p = c2^e - 1$ and $E/\mathbb{F}_p$ is on the *surface*

$$E[\mathbb{F}_p](2^\bullet) = \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2^{f-1}\mathbb{Z}$$

Translation

- There exists an $\mathbb{F}_p$-rational point P of order 2^{f-1} on E
- There exists an $\mathbb{F}_p$-rational point $\tilde{Q}$ of order 2^{f-1} on the twist E^t of E (twist is also on the surface)
- There exists a basis P, Q of $E(\mathbb{F}_{p^2})[2^{f-1}]$ with P_x, P_y, Q_x in $\mathbb{F}_p$ ($Q = \tau(\tilde{Q})$)

Slogan "Theta formulae of level 2 are x-only"

- The theta model gives an projective embedding of the Kummer of simple PPAVs
- Products of elliptic curves are not simple, so we get an embedding of $E_1/\pm \times E_2/\pm$ not $(E_1 \times E_2)/\pm$
- An isogeny chain looks like this $E_1/\pm \times E_2/\pm \to A_1/\pm \to \cdots \to A_n/\pm \to E_1'/\pm \times E_2'/\pm$
- $\ker(\Phi) = \langle (aP, cQ), (bP, dQ) \rangle \subseteq E \times E$ of can be represented by ($\mathbb{F}_p$-rational!) x-coordinates

In practice

- We still need y-coordinates to lift sign ambiguity in gluing (Superglue has fixed this problem)
- But for now we compute the diagonal isogenies and gluing over $\mathbb{F}_{p^2}$ ($\sim 1\%$)
- ...and then the remainder of the isogeny chain over $\mathbb{F}_p$ ($\sim 99\%$)
- ...our chains are long ($256 < 500$), this accounts for a very small amount of computation over $\mathbb{F}_{p^2}$

Implementation Results

Very rough comparison

		Lang.	512	1024	1500	2048	4096
Restricted	CSIDH†	C	26ms				
	SQALE*	C					5.75s**
	dCTIDH*	C				350ms**	
	This Work† (incl normeq)	Sage/C	0.160s	0.543s		3.564s	25.358s
	This Work† (excl normeq)	Sage	0.129s	0.324s		1.430s	8.100s
Unrestricted	SCALLOP*	C++	35s	750s			
	SCALLOP-HD*	Sage	88s	1140s			
	PEARL-SCALLOP*	C++	30s	58s	710s		
	KLaPoTi	Sage	207s				
		Rust	1.95s				
	PEGASIS	Sage	1.53s	4.21s		21.3s	121s
	qt-PEGASIS† (incl normeq)	Sage	0.808s	3.125s		12.189s	59.502s
	qt-PEGASIS† (excl normeq)	Sage	0.797s	2.905s		11.739s	61.186s

Table: †Measured on same hardware: AMD Ryzen 7 PRO 7840U@3.3GHz, Boost Disabled. *Converted from cycles to time @4GHz. The norm equation of **This Work** is implemented in C, and the ideal evaluation in Sage

Outlook

This is still work in progress

Security

- Better understanding of sampling distribution (apply well-known number-theoretic results)
- ...and more experimental data

Implementation

- Using better techniques for trial division in the norm equation (e.g. product trees)
- Low level C implementation (just begun with this)
- Prime-shape specific arithmetic (Scott's arithmetic)
- Assembly-optimised arithmetic (a la SQISign, can even reuse $\log(p) = 500$)
- AVX instructions to compute theta coordinates in parallel (promising recent work)

Constant timeness

- HD-dimensional isogenies are constant time friendly
- ...however we make a (very small!) random number of diagonal steps which (depends on the secret)
- Constant-time KeyGen (more precisely secret ideal sampling) looks tricky

Thank you for your attention

Slides https://rueg.re/isotum25