Efficient representation of isogenies

2023/07/10 — EWHA-KMS, Korea

Damien Robert

Équipe LFANT, Inria Bordeaux Sud-Ouest

Isogenies

- Elliptic curve: $E/k : y^2 = x^3 + ax + b$
- Algebraic group law

- Isogeny: $f : E_1 \to E_2$ with $f(0_{E_1}) = 0_{E_2}$
- $f(x,y) = \left(\frac{g(x)}{h(x)}, cy\left(\frac{g(x)}{h(x)}\right)'\right)$
- $f(P+Q) = f(P) + f(Q)$

- Kernel: $\operatorname{Ker} f = \{P \in E(\overline{k}) \mid f(P) = 0\}$
- Determines f, $\deg(f) = \# \operatorname{Ker} f$
- Conversely to every finite subgroup K corresponds an isogeny $f : E \to E/K$ with kernel K
- In this talk: separable isogenies (for simplicity)

- Isogeny evaluation: codomain E_2 and image of points $f(P)$

☺ Easy!

- Isogeny path: from (E_1, E_2) find an isogeny $f : E_1 \to E_2$

☹ Hard! Even for quantum computers!

$\Rightarrow$ Post quantum cryptosystems ☺

Isogeny path

Ordinary case:

- ☺ Commutative group action (from the class group of $\mathrm{End}(E)$)
- ☹ Quantum subexponential $L(1/2)$ algorithm (Kuperberg)

Supersingular case:

- Isogeny graph has good mixing properties
- Best algorithm is essentially exhaustive search (meet in the middle)
- ☺ Quantum exponential time
- ☹ No commutative group action

Isogeny evaluation

- Is isogeny evaluation actually easy?
- Depends on the representation and the degree N of f!

- Kernel equation: $K = \operatorname{Ker} f$ described by $h(x) = 0$
- Generator: $K = \langle T \rangle$
- For supersingular curves: ideal representation or suborder representation

- This talk: torsion representation
- Representation in polylog space with polylog time evaluation for any isogeny in any dimension

Kernel representation

- $K : h(x) = 0$ representing the kernel K of degree N
- $h(x) = \prod_{P \in K - 0_E} (x - x(P))$

- $f(x,y) = \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right)$
- $\frac{g(x)}{h(x)} = \#K.x - \sigma - u'(x) \frac{h'(x)}{h(x)} - 2u(x) \left(\frac{h'(x)}{h(x)} \right)'$ if $E : y^2 = u(x)$ [Kohel 1996]

- Space: polynomial of degee $O(N)$ over $\mathbb{F}_q$, so $O(N \log q)$ = linear space
- Evaluation: $O(N)$ arithmetic operations in $\mathbb{F}_q$ = linear time

Generator representation

- $K = \langle T \rangle$, K defined over $\mathbb{F}_q$, T defined over $\mathbb{F}_{q^d}$, $d = O(N)$
- $x(f(P)) = x(P) + \sum_{i=1}^{N-1} (x(P+iT) - x(iT))$ [Vélu 1971]
  $y(f(P)) = y(P) + \sum_{i=1}^{N-1} (y(P+iT) - y(iT))$
- Space: $O(1)$ elements over $\mathbb{F}_{q^d} = O(d \log q)$
- If $d = 1$ (or small): compact representation!
- Evaluation: $O(N)$ operations over $\mathbb{F}_{q^d} \Rightarrow$ linear if d small, quadratic if d large
- VeluSqrt [Bernstein, De Feo, Leroux, Smith 2020]: evaluation in $\widetilde{O}(\sqrt{N})$ over $\mathbb{F}_{q^d}$ (via a time/memory trade off)

Smooth degree

- If $N = \prod \ell_i^{e_i}$ is smooth, decompose f into a product $f = f_1 \circ f_2 \circ \ldots f_n$ of small degree isogenies ($n = O(\log N)$)
- Decomposed representation: complexity for evaluation depends on $\ell_N := \max(\ell_i)$

- Space: $O(n\ell_N \log q)$
- Evaluation: $O(n\ell_N \log q)$
- Logarithmic time!

Decomposing a smooth degree isogeny

- $K = \langle T \rangle$ of smooth degree $N = \prod_{i=1}^{m} \ell_i^{e_i}$
- Compute $S_1 = [N/\ell_1]T, f_1 : E \to E_1$ with kernel $K_1 = \langle S_1 \rangle$, and $T_1 = f_1(T)$
- Start again with E_1 and $K_1 = \langle T_1 \rangle$ of degree N/ℓ_1

- Complexity of the decomposition: $O(\log^2 N \ell_N)$ operations in $\mathbb{F}_{q^d}$
- Can be improved to $\widetilde{O}(\log N \ell_N)$ [De Feo, Jao, Plût 2011]

⚠ d can be large, $d = \Theta(N)$ in the worst case $\Rightarrow$ quasi-linear time

- CRT representation: $K = \prod_i K[\ell_i^{e_i}]$
- $K = \langle G_1, \dots, G_m \rangle$, with $K[\ell_i^{e_i}] = \langle G_i \rangle$
- Cost depends on the degrees d_i of the field of definition of each G_i: $\widetilde{O}(m(\sum e_i)\ell_N)$ operations in fields $\mathbb{F}_{q^{d_i \vee d_j}}$

$\Rightarrow$ Polynomial time in ℓ_N and the d_i (and $\log N, \log q$)

- Example: the cost of decomposing a 2^n-isogeny depends on whether $E[2^n]$ lies over a small or big extension
- If N B-powersmooth, $d_i = O(B)$

$\Rightarrow$ Decomposition in polylogarithmic time from a CRT representation of K for powersmooth N

Decomposing a smooth degree isogeny from a kernel equation

- $N = \ell_1 N_1$, kernel equation: $h(x) = 0$
- Work with the formal point $P = (x, y)$ in the algebra $\mathcal{A} = \mathbb{F}_q[x, y]/(h(x), y^2 - x^3 - ab - b)$
- The denominator of $\ell_1.P$ gives an equation $g(x) = 0$ for $K[\ell_1]$, $g \mid h$
- This allows to compute $f_1(P)$, with $\operatorname{Ker} f_1 = K[\ell_1]$ via $O(\ell_1)$ operations in $\mathcal{A}$
- Iterate

- Decomposition cost: $\widetilde{O}(\ell_N N)$ arithmetic operations

Ideal and suborder representation

- For supersingular curves: Deuring correspondance
- $\mathrm{End}(E)$ = quaternion algebra
- Ideals = Isogenies

☺ Ideal representation: evaluation in polylogarithmic time!

☹ Leaks too much informations: the isogeny path becomes trivial

- Variant: suborder representation [Leroux 2022], leaks less informations

Summary

- Kernel representation: linear space and time
- Generator representation: possibly compact, linear or quadratic time
- If N smooth: decomposed representation = logarithmic space and time
- Decomposition cost given a CRT representation $K = \langle G_1, \dots, G_m \rangle$: polynomial time in $d = \max(d_i)$ and $\ell_N = \max(\ell \mid N)$; so polylogarithmic time if N powersmooth

- What if N is a large prime?
- Ideal representation or suborder representation

☹ Leaks information, only available for supersingular curves

Scalar multiplication

- Scalar multiplication: $P \mapsto N.P$
- Double and add: $O(\log N)$ arithmetic operations, even if N is prime!

- $F : E^2 \to E^2, (P_1, P_2) \mapsto (P_1 + P_2, P_1 - P_2)$ is a 2-isogeny in dimension 2.
- $F = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$
- Double: $F(T,T) = (2T, 0)$.
- Add: $F(T,Q) = (T+Q, T-Q)$.

- We can evaluate $N.P$ as a composition of $O(\log N)$ evaluations of F, projections $E^2 \to E$ and embeddings $E \to E^2$.
- Double and add on E = 2-isogenies in dimension 2

The embedding lemma [R. 2022]

- A N-isogeny $f : A \to B$ in dimension g can always be efficiently embedded into a N' isogeny $F : A' \to B'$ in dimension $8g$ (and sometimes $4g, 2g$) for any $N' \geq N$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, hook] & B \\ A' \arrow[r, "F"] & B' \arrow[u, two heads] \end{tikzcd}$$

- Considerable flexibility (at the cost of going up in dimension).
- Breaks SIDH ([Castryck-Decru 2022], [Maino-Martindale 2022] in dimension 2, [R. 2022] in dimension 4 or 8)

- Write $N' - N = a_1^2 + a_2^2 + a_3^2 + a_4^2$
- $F = \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 & \hat{f} & 0 & 0 & 0 \\ a_2 & a_1 & a_4 & -a_3 & 0 & \hat{f} & 0 & 0 \\ a_3 & -a_4 & a_1 & a_2 & 0 & 0 & \hat{f} & 0 \\ a_4 & a_3 & -a_2 & a_1 & 0 & 0 & 0 & \hat{f} \\ -f & 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 \\ 0 & -f & 0 & 0 & -a_2 & a_1 & -a_4 & a_3 \\ 0 & 0 & -f & 0 & -a_3 & a_4 & a_1 & a_2 \\ 0 & 0 & 0 & -f & -a_4 & -a_3 & a_2 & a_1 \end{pmatrix}$

The embedding lemma for isogeny representation

- To embed the N-isogeny f into the N'-isogeny F, needs $(P, Q, f(P), f(Q))$ for (P, Q) a basis of $E[N']$

⇒ Torsion representation

- Evaluation: evaluate an N'-isogeny F in higher dimension g: $O(N'^g)$ [Lucicz-R. 2008–2022] (Warning: the $O()$ hides a constant 2^g)
- N' smooth: decompose F into small isogenies

⇒ Evaluation in polylogarithmic time

- Decomposition: use a CRT basis of $E[N']$

⇒ Decomposition in polylogarithmic time if $d = \max(d_i)$ small, e.g. N' powersmooth.

The torsion representation

- Represent f by its degree N and
$$(P_i, Q_i, f(P_i), f(Q_i))$$
for (P_i, Q_i) a basis of $E[N_i]$, with N_i coprimes and prime to N
- Needs $N' = \prod N_i > N$
Trick via the dual isogeny: only needs $\prod N_i > \sqrt{N}$ (for uniqueness: $\prod N_i^2 > 4N$)
- Evaluation (and decomposition) polynomial in the ℓ_{N_i} and d_i, the degrees of the fields of definition of (P_i, Q_i)

☺ Evaluation in polylogarithmic time (take small N_i)

☺ Even faster if $E[\ell^m]$ rational for large m and small ℓ:
if $E[2^m]$ is rational embed any degree $N < 2^m$ isogeny into a 2^m-isogeny!
(Dual isogeny trick: embed any degree $N < 2^{2m}$ isogeny into two 2^m-isogenies)

☺ Universal: can be efficiently recovered from any other efficient representation

☺ Works in any dimension (ie on abelian varieties)

☹ Needs the codomain of f (at least implicitly: it can be recovered from the $f(P_i), f(Q_i)$)

Using the torsion representation

- Dual: If $f : E \to E'$ has an efficient representation, its dual $\hat{f} : E' \to E$ too
- Since $\hat{f}(f(P)) = NP$, the torsion representation is $(f(P_i), f(Q_i), NP_i, NQ_i)$ on $E'[N_i]$
- Example: Frobenius π_p has efficient evaluation $(x,y) \mapsto (x^p, y^p)$, hence the Verschiebung $\hat{\pi}_p$ too on any point

- Division: If $f = mg$ has an efficient representation, g too
- Torsion representation: $(P_i, Q_i, f(P_i)/m, f(Q_i)/m)$ on $E[N_i]$, N_i prime to m
  (if necessary compute new evaluation points for f)
- Allows to evaluate g on $E[m]$
- Example: if $A/\mathbb{F}_q$ ordinary, $\alpha \in \mathrm{End}(E) = (a_0 + a_1\pi + \cdots + a_g\pi^g)/m$, gives an efficient representation for any endomorphism

Using the torsion representation

- Addition: if f, g have an efficient representation, $f+g$ too

⚠ Needs $\deg(f+g)$

- $[\deg(f+g)] = \widehat{f+g} \circ (f+g)$ can be computed on $E[N_i]$ from evaluating $f, g, \hat{f}, \hat{g}$; this gives $\deg(f+g) \mod N_i$
- Recover $\deg(f+g)$ by the CRT and the Cauchy-Schwarz bound:

$$\deg(f+g) \leq \deg(f) + \deg(g) + 2\sqrt{\deg(f)\deg(f)}$$

- Torsion representation: $(P_i, Q_i, f(P_i) + g(P_i), f(Q_i) + g(Q_i))$ on $E[N_i]$
  (if necessary compute new evaluation points for f, g)

- Lifting: Lift f by lifting F
  (Lift to $\mathbb{Q}_q/p^m$ or to $\mathbb{F}_q[[\epsilon]]/\epsilon^m$)

Algorithmic applications [R. 2022]

- $E/\mathbb{F}_q$ ordinary elliptic curve, $K = \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$. Given the factorisation of $[O_K : \mathbb{Z}[\pi]]$, compute $\mathrm{End}(E)$ in polynomial time (via efficient division).
  Factorisation: quantum polynomial time, classical subexponential time
- Previously: no quantum polynomial time algorithm known
  Classical algorithm in $L(1/2)$ under GRH [Bisson–Sutherland 2009]

- Compute the canonical lift $\hat{E}/\mathbb{Z}_q$ in polynomial time
- Previously: $L(1/2)$ under GRH [Couveignes–Henocq 2002]

- Compute the modular polynomial Φ_ℓ in quasi-linear time in any dimension g
- Previously: no algorithm known to compute Φ_ℓ in quasi-linear time when $g > 2$

Point counting and canonical lifts

$E/\mathbb{F}_q, q = p^n$

- [Schoof 1985]: $\widetilde{O}(n^5 \log^5 p)$ (Étale cohomology)
- [SEA 1992]: $\widetilde{O}(n^4 \log^4 p)$ (Heuristic)

- [Kedlaya 2001]: $\widetilde{O}(n^3 p)$ (Rigid cohomology)
- [Harvey 2007]: $\widetilde{O}(n^{3.5} p^{1/2} + n^5 \log p)$

- [Satoh 2000] (canonical lifts of ordinary curves): $\widetilde{O}(n^2 p^2)$ (Crystalline cohomology)
- [Maiga – R. 2021]: $\widetilde{O}(n^2 p)$
- [R. 2022]: $\widetilde{O}(n^2 \log^8 p + n \log^{11} p)$

Cryptographic applications

- Free protocols from the shackle of using only smooth degree isogenies
- Choose E with large rational 2^m-torsion $\Rightarrow$ embed N-isogenies into higher dimensional 2^m-isogenies

- SQISignHD [Dartois, Leroux, R., Wesolowski 2023]: post-quantum signature scheme
- Signing in dimension 1, verification in dimension 4
- Public key: 64B, Signature: 105B
  Prior Art: SQISign: 204B, Lattices: 666B–2420B, (ECDSA: 64B)

- FESTA [Basso, Maino, Pope 2023]: encryption in dimension 1, decryption in dimension 2
- Identity based encryption [Fouotsa 2023]: use dimension 8

Open problems

- Optimize 2^m-isogenies in higher dimension
- Work in progress to optimize the constants:

g	Old ratios	New ratios
2	×6	×4
4	×160	×32
8	×75000	×1024

- Main drawback of the torsion representation: needs enough evaluation points
- Efficient conversion from the generator representation for a prime degree isogeny?
- Efficient representation which does not need the codomain?