Solving Endrings with low memory via isogeny ladders and small orientations

Alessandro Sferlazza
joint work with Lorenz Panny, Ryan Rueger, Aleksei Udovenko

Technical University of Munich

Wednesday, 12 August 2026
AM-PQC Workshop, Ohrid

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)

[1] more than four weeks!

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)
2. Find an isogeny over $\mathbb{F}_p$ between E_1 and a special target curve E_0 of known End ring.
   $E_1 \rightsquigarrow E_0$ (easier: class group action vectorization, takes $\widetilde{O}(p^{1/4})$)

[1] more than four weeks!

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)
2. Find an isogeny over $\mathbb{F}_p$ between E_1 and a special target curve E_0 of known End ring.
   $E_1 \rightsquigarrow E_0$ (easier: class group action vectorization, takes $\widetilde{O}(p^{1/4})$)
3. Transfer End ring knowledge $E_0 \rightsquigarrow E_1 \rightsquigarrow E$.

[1] more than four weeks!

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)
2. Find an isogeny over $\mathbb{F}_p$ between E_1 and a special target curve E_0 of known End ring.
   $E_1 \rightsquigarrow E_0$ (easier: class group action vectorization, takes $\widetilde{O}(p^{1/4})$)
3. Transfer End ring knowledge $E_0 \rightsquigarrow E_1 \rightsquigarrow E$.

$\rightsquigarrow$ Easily parallelizable and low-memory ✓

[1] more than four weeks!

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)
2. Find an isogeny over $\mathbb{F}_p$ between E_1 and a special target curve E_0 of known End ring.
   $E_1 \rightsquigarrow E_0$ (easier: class group action vectorization, takes $\widetilde{O}(p^{1/4})$)
3. Transfer End ring knowledge $E_0 \rightsquigarrow E_1 \rightsquigarrow E$.

$\rightsquigarrow$ Easily parallelizable and low-memory ✓

$\rightsquigarrow$ Easily customizable approach:

- Shape of random walk: linear walk in an ℓ-graph, depth-first search in ℓ-isogeny tree, ...

[1] more than four weeks!

Finding endomorphism rings: quite hard, right?

Problem: Given $E/\mathbb{F}_{p^2} \in \mathcal{S}_{p^2}$ supersingular elliptic curve, find explicit description of $\mathrm{End}(E)$.

Old[1]-style solution: Delfs–Galbraith approach:

1. Walk randomly in the supersingular isogeny graph until we find a curve defined over $\mathbb{F}_p$:
   $E \rightsquigarrow E_1 \in \mathcal{S}_p$ (hard, takes $\widetilde{O}(\#\mathcal{S}_{p^2}/\#\mathcal{S}_p)) \subseteq \widetilde{O}(p^{1/2})$)
2. Find an isogeny over $\mathbb{F}_p$ between E_1 and a special target curve E_0 of known End ring.
   $E_1 \rightsquigarrow E_0$ (easier: class group action vectorization, takes $\widetilde{O}(p^{1/4})$)
3. Transfer End ring knowledge $E_0 \rightsquigarrow E_1 \rightsquigarrow E$.

$\rightsquigarrow$ Easily parallelizable and low-memory ✓

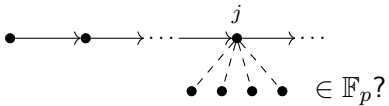
$\rightsquigarrow$ Easily customizable approach:

- Shape of random walk: linear walk in an ℓ-graph, depth-first search in ℓ-isogeny tree, ...
- Choice of target subgraph: replace $\mathcal{S}_p$ with other special subgraphs...

[1] more than four weeks!

Looking for oriented curves

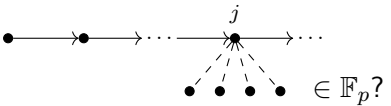
Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.

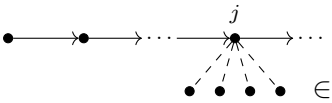


How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



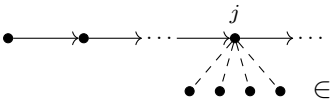
How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

Note: the neighbours of j are $\mathbb{F}_p$-nodes, aka $\mathbb{Z}[\sqrt{-p}]$-oriented $\Leftrightarrow$ j is oriented by $\mathbb{Z}[\ell\sqrt{-p}]$, with def: E is $\mathcal{O}$-oriented $\Leftrightarrow$ $\mathcal{O} \hookrightarrow \mathrm{End}(E)$.

[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

Note: the neighbours of j are $\mathbb{F}_p$-nodes, aka $\mathbb{Z}[\sqrt{-p}]$-oriented $\Leftrightarrow$ j is oriented by $\mathbb{Z}[\ell\sqrt{-p}]$, with def: E is $\mathcal{O}$-oriented $\Leftrightarrow$ $\mathcal{O} \hookrightarrow \mathrm{End}(E)$.

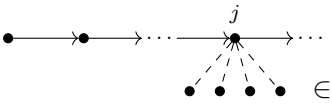
Extend this idea: Fix a set of quadratic rings, say $\mathcal{O}_d = \mathbb{Z}[\sqrt{-dp}]$ for several d.
For each d, fix $E_{0,d}$ that is $\mathcal{O}_d$-orientable and with known End ring.

- Random-walk until we hit an $\mathcal{O}_d$-orientable curve E_1 for *any* d not fixed a priori.

[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

Note: the neighbours of j are $\mathbb{F}_p$-nodes, aka $\mathbb{Z}[\sqrt{-p}]$-oriented $\Leftrightarrow$ j is oriented by $\mathbb{Z}[\ell\sqrt{-p}]$, with def: E is $\mathcal{O}$-oriented $\Leftrightarrow$ $\mathcal{O} \hookrightarrow \mathrm{End}(E)$.

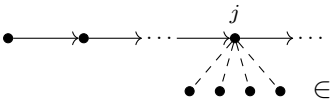
Extend this idea: Fix a set of quadratic rings, say $\mathcal{O}_d = \mathbb{Z}[\sqrt{-dp}]$ for several d.
For each d, fix $E_{0,d}$ that is $\mathcal{O}_d$-orientable and with known End ring.

- Random-walk until we hit an $\mathcal{O}_d$-orientable curve E_1 for *any* d not fixed a priori.
  speedup: ▷ wider choice of target curves & shared computation for different d

[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

Note: the neighbours of j are $\mathbb{F}_p$-nodes, aka $\mathbb{Z}[\sqrt{-p}]$-oriented $\Leftrightarrow$ j is oriented by $\mathbb{Z}[\ell\sqrt{-p}]$, with def: E is $\mathcal{O}$-oriented $\Leftrightarrow$ $\mathcal{O} \hookrightarrow \mathrm{End}(E)$.

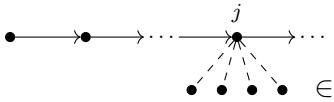
Extend this idea: Fix a set of quadratic rings, say $\mathcal{O}_d = \mathbb{Z}[\sqrt{-dp}]$ for several d.
For each d, fix $E_{0,d}$ that is $\mathcal{O}_d$-orientable and with known End ring.

- Random-walk until we hit an $\mathcal{O}_d$-orientable curve E_1 for *any* d not fixed a priori.
  speedup: ▷ wider choice of target curves & shared computation for different d
- Solve $\mathcal{O}_d$-vectorization: find $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O}_d)$ with $[\mathfrak{a}] \star E_1 = E_{0,d}$.

[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488

Looking for oriented curves

Neighbourhood inspection from SuperSolver:[2] check whether any ℓ-neighbours are over $\mathbb{F}_p$.



How? Check for common roots of $\Phi_\ell(j, Y)$ and its Galois-conjugate $\Phi_\ell(j^p, Y)$ (compute the gcd).

Note: the neighbours of j are $\mathbb{F}_p$-nodes, aka $\mathbb{Z}[\sqrt{-p}]$-oriented $\Leftrightarrow$ j is oriented by $\mathbb{Z}[\ell\sqrt{-p}]$, with def: E is $\mathcal{O}$-oriented $\Leftrightarrow$ $\mathcal{O} \hookrightarrow \mathrm{End}(E)$.

Extend this idea: Fix a set of quadratic rings, say $\mathcal{O}_d = \mathbb{Z}[\sqrt{-dp}]$ for several d.
For each d, fix $E_{0,d}$ that is $\mathcal{O}_d$-orientable and with known End ring.

- Random-walk until we hit an $\mathcal{O}_d$-orientable curve E_1 for *any* d not fixed a priori.
  speedup: ▷ wider choice of target curves & shared computation for different d
- Solve $\mathcal{O}_d$-vectorization: find $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O}_d)$ with $[\mathfrak{a}] \star E_1 = E_{0,d}$.

How to check orientability? For $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$ $\rightsquigarrow$ test if E, E^p are d-isogenous.
$\mathcal{O} = \mathbb{Z}[\ell_1\sqrt{-\ell_2 p}]$ $\rightsquigarrow$ can use gcd and resultants, generalizing SuperSolver.[3]

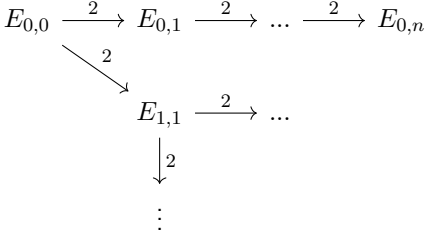
[2] Corte-Real Santos, Costello, Shi. Eprint 2021/1488
[3] Corte-Real Santos, Herlédan Le Merdy, Macula, Meyer, Morrison, Orvis. Eprint 2026/1219

Isogeny ladders

$$E_{0,0} \xrightarrow{2} E_{0,1} \xrightarrow{2} \dots \xrightarrow{2} E_{0,n}$$

- Linear walk: each 2-isogeny arrow costs a sqrt.

Isogeny ladders

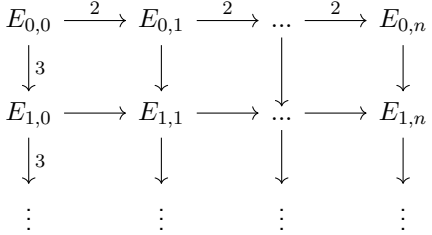


- ▷ Linear walk: each 2-isogeny arrow costs a sqrt.
- ▷ Walk in a 2-isogeny tree:[3]

$\rightsquigarrow$ each sqrt gives 2 child nodes.

[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

Isogeny ladders

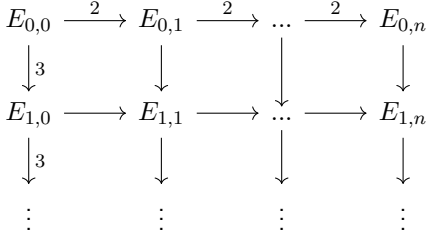


- ▷ Linear walk: each 2-isogeny arrow costs a sqrt.
- ▷ Walk in a 2-isogeny tree:[3]

  $\rightsquigarrow$ each sqrt gives 2 child nodes.
- ▷ Our solution: explore a 2D grid, using $\ell_2 = 3$ for vertical lines $\rightsquigarrow$ SIDH ladders :D

[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

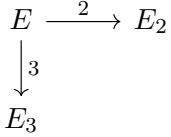
Isogeny ladders



- ▷ Linear walk: each 2-isogeny arrow costs a sqrt.
- ▷ Walk in a 2-isogeny tree:[3]

$\rightsquigarrow$ each sqrt gives 2 child nodes.

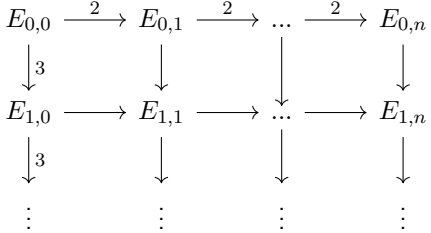
- ▷ Our solution: explore a 2D grid, using $\ell_2 = 3$ for vertical lines $\rightsquigarrow$ SIDH ladders :D



By construction, we know E_2, E_3 are 6-isogenous.

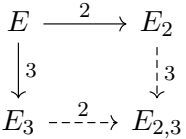
[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

Isogeny ladders



- ▷ Linear walk: each 2-isogeny arrow costs a sqrt.
- ▷ Walk in a 2-isogeny tree:[3]

  ⇝ each sqrt gives 2 child nodes.
- ▷ Our solution: explore a 2D grid, using $\ell_2 = 3$ for vertical lines ⇝ SIDH ladders :D

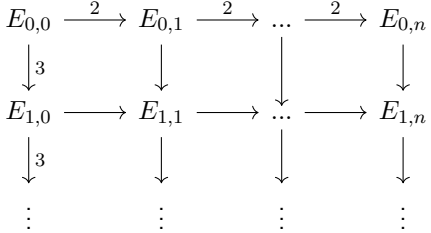


By construction, we know E_2, E_3 are 6-isogenous.

⇝ A 3-neighbour of E_2 will be a 2-neighbour of E_3.

[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

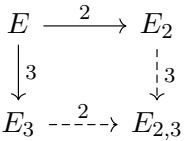
Isogeny ladders



▷ Linear walk: each 2-isogeny arrow costs a sqrt.

▷ Walk in a 2-isogeny tree:[3]
⇝ each sqrt gives 2 child nodes.

▷ Our solution: explore a 2D grid, using $\ell_2 = 3$ for vertical lines ⇝ SIDH ladders :D

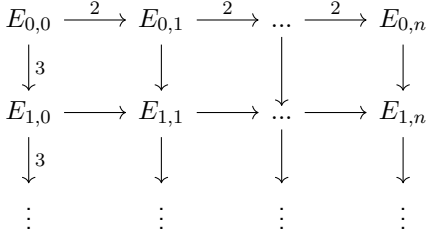


By construction, we know E_2, E_3 are 6-isogenous.
⇝ A 3-neighbour of E_2 will be a 2-neighbour of E_3.

Look for common roots $j(E_{2,3})$ of $\Phi_3(j(E_2), Y)$ and $\Phi_2(j(E_3), Y)$.
⇝ find $E_{2,3}$ with just a gcd of polynomials ✓

[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

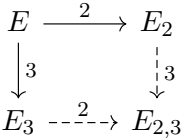
Isogeny ladders



▷ Linear walk: each 2-isogeny arrow costs a sqrt.

▷ Walk in a 2-isogeny tree:[3]
⇝ each sqrt gives 2 child nodes.

▷ Our solution: explore a 2D grid, using $\ell_2 = 3$ for vertical lines ⇝ SIDH ladders :D



By construction, we know E_2, E_3 are 6-isogenous.
⇝ A 3-neighbour of E_2 will be a 2-neighbour of E_3.

Look for common roots $j(E_{2,3})$ of $\Phi_3(j(E_2), Y)$ and $\Phi_2(j(E_3), Y)$.
⇝ find $E_{2,3}$ with just a gcd of polynomials ✓

✓ Square roots required only for the outer sides of the grid
⇝ sqrts amortized out, asymptotic $\log(p)$-speedup

[3] Chi-Dominguez. Eprint 2025/226 (SuperSearcher)

Graph connectivity vs success probability: 1/2

When random-walking, what's the probability of hitting a nice curve?

Graph connectivity vs success probability: 1/2

When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

Graph connectivity vs success probability: 1/2

When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

⚠ Only true up to a constant :(

- We don't visit independent random curves, but rather lines/trees/grids of isogenous ones

Graph connectivity vs success probability: 1/2

When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

⚠ Only true up to a constant :(

- We don't visit independent random curves, but rather lines/trees/grids of isogenous ones
- The target subgraph is not random: the way it is connected interacts our exploration.

Graph connectivity vs success probability: 1/2

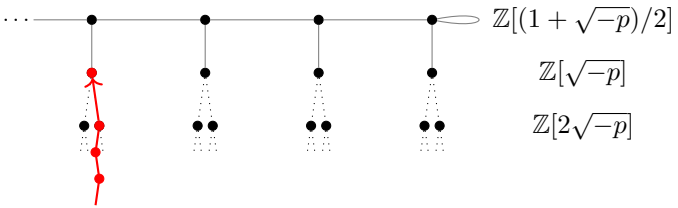
When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

⚠ Only true up to a constant :(

- We don't visit independent random curves, but rather lines/trees/grids of isogenous ones
- The target subgraph is not random: the way it is connected interacts our exploration.

Example: curves over $\mathbb{F}_p$. When considering orientability with respect to orders in $\mathbb{Q}(\sqrt{-p})$, the supersingular isogeny graph *mostly* looks like a folded volcano:



Graph connectivity vs success probability: 1/2

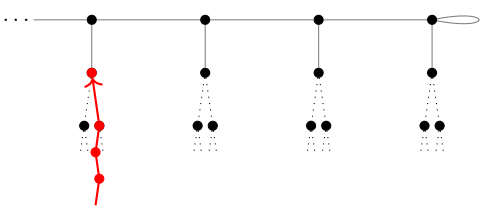
When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

⚠ Only true up to a constant :(

- We don't visit independent random curves, but rather lines/trees/grids of isogenous ones
- The target subgraph is not random: the way it is connected interacts our exploration.

Example: curves over $\mathbb{F}_p$. When considering orientability with respect to orders in $\mathbb{Q}(\sqrt{-p})$, the supersingular isogeny graph *mostly* looks like a folded volcano:



💡 Paths coming from outside $\mathbb{F}_p$ will first reach a curve on the floor via an ascending isogeny.

Graph connectivity vs success probability: 1/2

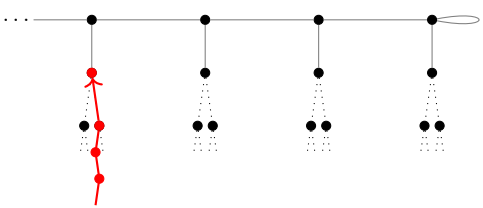
When random-walking, what's the probability of hitting a nice curve?

First guess: we visit random nodes $\rightsquigarrow$ success prob $\approx \#(\text{nice curves})/\#(\text{all curves})$.

⚠ Only true up to a constant :(

- We don't visit independent random curves, but rather lines/trees/grids of isogenous ones
- The target subgraph is not random: the way it is connected interacts our exploration.

Example: curves over $\mathbb{F}_p$. When considering orientability with respect to orders in $\mathbb{Q}(\sqrt{-p})$, the supersingular isogeny graph *mostly* looks like a folded volcano:



💡 Paths coming from outside $\mathbb{F}_p$ will first reach a curve on the floor via an ascending isogeny.

$\rightsquigarrow$ only part of the target subgraph is "visible" from outside. ⚠

Graph connectivity vs success probability: 2/2

The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;

Graph connectivity vs success probability: 2/2

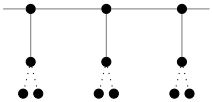
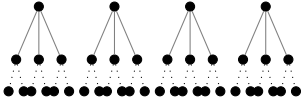
The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;
- $\ell = 2$ is more complicated: quadratic residuosity; $\mathcal{O}$ can be maximal or of conductor 2.

Graph connectivity vs success probability: 2/2

The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

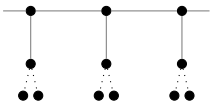
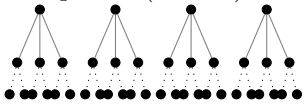
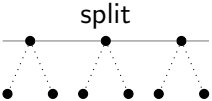
- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;
- $\ell = 2$ is more complicated: quadratic residuosity; $\mathcal{O}$ can be maximal or of conductor 2.

	$p \equiv 7 \pmod 8$	$p \equiv 3 \pmod 8$	$p \equiv 1 \pmod 4$
$\ell = 2$			

Graph connectivity vs success probability: 2/2

The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

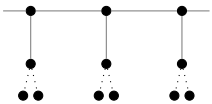
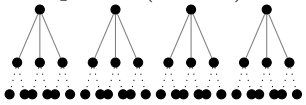
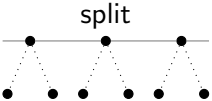
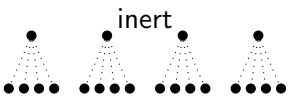
- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;
- $\ell = 2$ is more complicated: quadratic residuosity; $\mathcal{O}$ can be maximal or of conductor 2.

	$p \equiv 7 \pmod 8$	$p \equiv 3 \pmod 8$	$p \equiv 1 \pmod 4$
$\ell = 2$			
ℓ odd	split 	ramified 	inert 

Graph connectivity vs success probability: 2/2

The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;
- $\ell = 2$ is more complicated: quadratic residuosity; $\mathcal{O}$ can be maximal or of conductor 2.

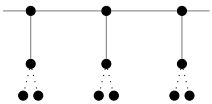
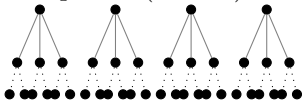
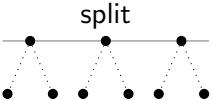
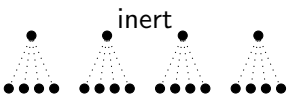
	$p \equiv 7 \pmod 8$	$p \equiv 3 \pmod 8$	$p \equiv 1 \pmod 4$
$\ell = 2$			
ℓ odd	split 	ramified 	inert 

 Less horizontal isogenies $\leftrightarrow$ more floor nodes, target graph more scattered in $\mathbb{F}_{p^2}$ graph.

Graph connectivity vs success probability: 2/2

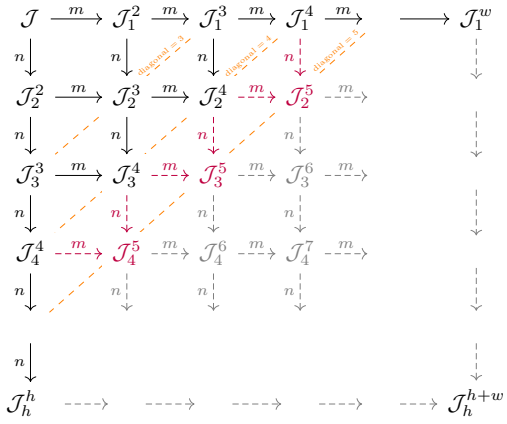
The shape of the $\mathbb{Z}[\sqrt{-dp}]$-oriented ℓ-isogeny graph depends on

- Whether (ℓ) is split, ramified or inert in $\mathcal{O} = \mathbb{Z}[\sqrt{-dp}]$;
- $\ell = 2$ is more complicated: quadratic residuosity; $\mathcal{O}$ can be maximal or of conductor 2.

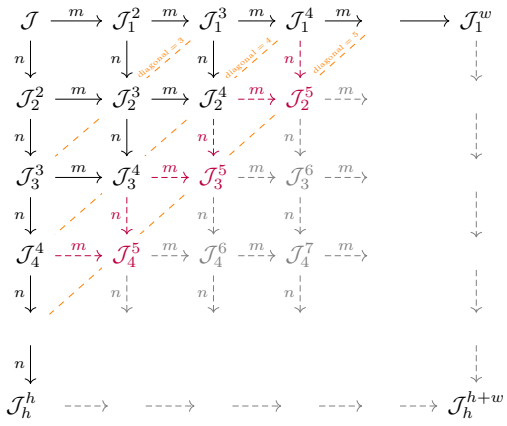
	$p \equiv 7 \pmod 8$	$p \equiv 3 \pmod 8$	$p \equiv 1 \pmod 4$
$\ell = 2$			
ℓ odd	split 	ramified 	inert 

- Less horizontal isogenies $\leftrightarrow$ more floor nodes, target graph more scattered in $\mathbb{F}_{p^2}$ graph.
- In $(2, 3)$-isogeny ladders, the (negative) effect of both primes $\ell = 2, 3$ combines.

Implementation remarks

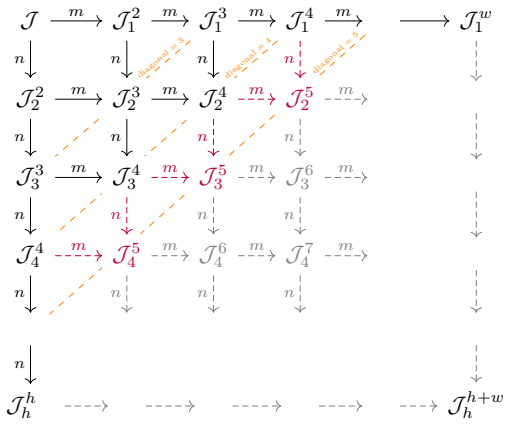


Implementation remarks



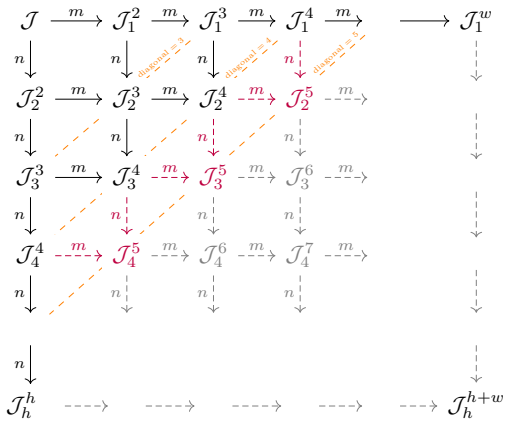
- Ladder square computations (aka pushout) can be parallelized along diagonals

Implementation remarks



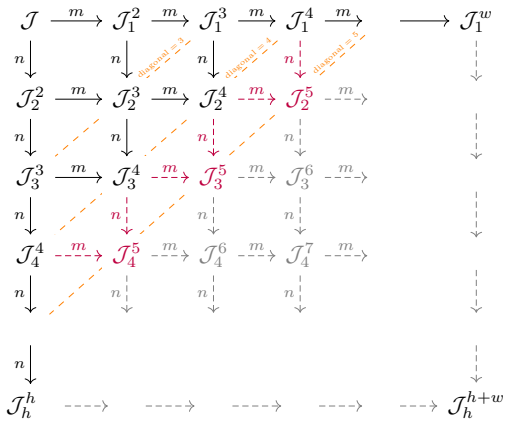
- Ladder square computations (aka pushout) can be parallelized along diagonals
- Low memory impact: $O(\log p)$-sized h, w & deterministic memory access pattern

Implementation remarks



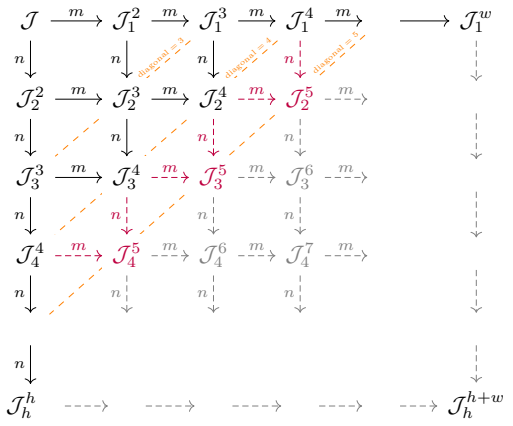
- Ladder square computations (aka pushout) can be parallelized along diagonals
- Low memory impact: $O(\log p)$-sized h, w & deterministic memory access pattern
- Pushouts are very cheap, so that checking large orientations does not reward:
  $\rightsquigarrow$ enough to use simple formulas with $\ell_i \in 2, 3$.

Implementation remarks



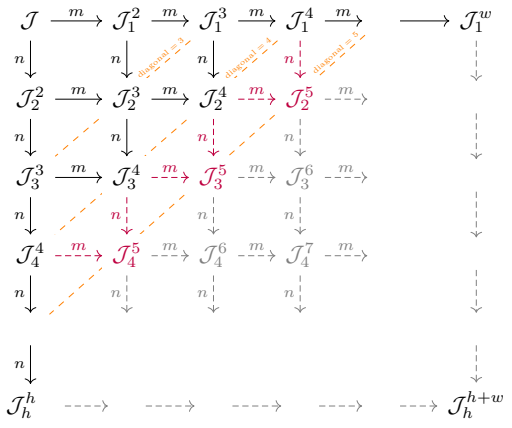
- Ladder square computations (aka pushout) can be parallelized along diagonals
- Low memory impact: $O(\log p)$-sized h, w & deterministic memory access pattern
- Pushouts are very cheap, so that checking large orientations does not reward:
  $\rightsquigarrow$ enough to use simple formulas with $\ell_i \in 2, 3$.
- Pushouts help in case we need to compute *many* $n^a m^b d$-isogenies with small n, m. (vectorization, Wesolowski's $p^{1/3}$-attack)

Implementation remarks



- Ladder square computations (aka pushout) can be parallelized along diagonals
- Low memory impact: $O(\log p)$-sized h, w & deterministic memory access pattern
- Pushouts are very cheap, so that checking large orientations does not reward:
  $\rightsquigarrow$ enough to use simple formulas with $\ell_i \in 2, 3$.
- Pushouts help in case we need to compute *many* $n^a m^b d$-isogenies with small n, m. (vectorization, Wesolowski's $p^{1/3}$-attack)
- Our record: 100 bit prime in ∼100 GPU hours, ∼16 GB of RAM

Implementation remarks



- Ladder square computations (aka pushout) can be parallelized along diagonals
- Low memory impact: $O(\log p)$-sized h, w & deterministic memory access pattern
- Pushouts are very cheap, so that checking large orientations does not reward:
  $\rightsquigarrow$ enough to use simple formulas with $\ell_i \in 2, 3$.
- Pushouts help in case we need to compute *many* $n^a m^b d$-isogenies with small n, m. (vectorization, Wesolowski's $p^{1/3}$-attack)
- Our record: 100 bit prime in ~100 GPU hours, ~16 GB of RAM

Thank you! Questions?