Disorientation faults in CSIDH

Tanja Lange
(with lots of slides by Chloe Martindale and Lorenz Panny)

Eindhoven University of Technology

18 October 2022

Isogenies

An *isogeny* of elliptic curves is a non-zero map $E \to E'$

- given by *rational functions*
- that is a *group homomorphism*.

The *degree* of a separable isogeny is the size of its *kernel*.

Isogenies

An *isogeny* of elliptic curves is a non-zero map $E \to E'$

- given by *rational functions*
- that is a *group homomorphism*.

The *degree* of a separable isogeny is the size of its *kernel*.

Example #1: For each $m \neq 0$, the *multiplication-by-m map*

$$[m]\colon E \to E$$

is an isogeny from E to itself.

If $m \neq 0$ in the base field, its kernel is

$$E[m] \;\cong\; \mathbb{Z}/m \times \mathbb{Z}/m.$$

Thus $[m]$ is a degree-m^2 isogeny.

Isogenies

An *isogeny* of elliptic curves is a non-zero map $E \to E'$

- given by *rational functions*
- that is a *group homomorphism*.

The *degree* of a separable isogeny is the size of its *kernel*.

Example #2: For any a and b, the map $\iota\colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$

defines a degree-1 isogeny of the elliptic curves

$$\{y^2 = x^3 + ax + b\} \longrightarrow \{y^2 = x^3 + ax - b\}\,.$$

It is an *isomorphism*; its kernel is $\{\infty\}$.

Isogenies

An *isogeny* of elliptic curves is a non-zero map $E \to E'$

- given by *rational functions*
- that is a *group homomorphism*.

The *degree* of a separable isogeny is the size of its *kernel*.

Example #3:

$$(x, y) \mapsto \left(\frac{x^3-4x^2+30x-12}{(x-2)^2}, \frac{x^3-6x^2-14x+35}{(x-2)^3} \cdot y \right)$$

defines a degree-3 isogeny of the elliptic curves

$$\{y^2 = x^3 + x\} \longrightarrow \{y^2 = x^3 - 3x + 3\}$$

over $\mathbb{F}_{71}$. Its kernel is $\{(2, 9), (2, -9), \infty\}$.

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)
- We can *walk efficiently* on these graphs.

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)
- We can *walk efficiently* on these graphs.
- *Fast mixing*: short paths to (almost) all nodes.

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)
- We can *walk efficiently* on these graphs.
- *Fast mixing*: short paths to (almost) all nodes.
- *No efficient* algorithms to *recover paths* from endpoints.
  (*Both* classical and quantum!)

Big picture

- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)
- We can *walk efficiently* on these graphs.
- *Fast mixing*: short paths to (almost) all nodes.
- *No efficient* algorithms to *recover paths* from endpoints.
  (*Both* classical and quantum!)
- *Enough structure* to *navigate* the graph meaningfully.
  That is: some *well-behaved* "directions" to describe paths. More later.

Big picture

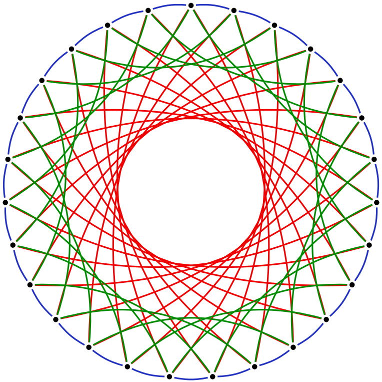
- Isogenies are a source of *exponentially*-sized *graphs*.
- Isogeny graph: Nodes are isomorphism classes of curves, edges are isogenies.

  (We usually care about *sub*graphs with certain properties.)
- We can *walk efficiently* on these graphs.
- *Fast mixing*: short paths to (almost) all nodes.
- *No efficient* algorithms to *recover paths* from endpoints.
  (*Both* classical and quantum!)
- *Enough structure* to *navigate* the graph meaningfully.
  That is: some *well-behaved* "directions" to describe paths. More later.

It is easy to construct graphs that satisfy *almost* all of these
"Almost" is not good enough for crypto!

Different isogeny graphs

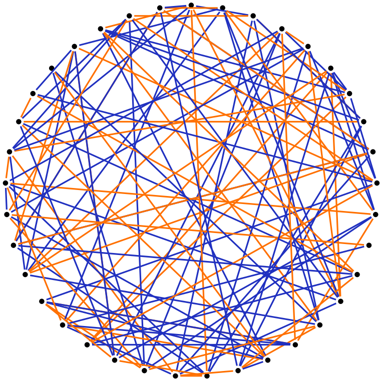
There are two distinct families of systems:



$q = p$

CSIDH [ˈsiːˌsaɪd]

https://csidh.isogeny.org



$q = p^2$

SIDH

https://sike.org

CSIDH ['siːˌsaɪd]

(Castryck, Lange, Martindale, Panny, Renes; 2018)

Why CSIDH?

- Closest thing we have in PQC to normal DH key exchange:
  Keys can be reused, blinded; no difference between initiator &responder.
- Public keys are represented by some $A \in \mathbb{F}_p$; p fixed prime.
- Alice computes and distributes her public key A.
  Bob computes and distributes his public key B.
- Alice and Bob do computations on each other's public keys
  to obtain shared secret.
- Fancy math: computations start on some elliptic curve $E_A : y^2 = x^3 + Ax^2 + x$, use
  isogenies to move to a different curve.
- Computations need arithmetic (add, mult, div) modulo p and
  elliptic-curve computations.

CSIDH in one slide

- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.

CSIDH in one slide

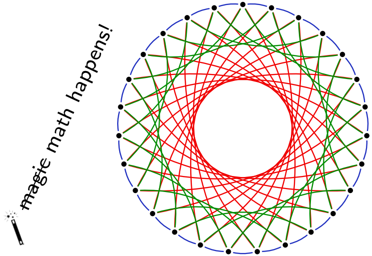
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p+1$ points$\}$.

CSIDH in one slide

- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p+1$ points$\}$.
- Look at the ℓ_i-isogenies defined over $\mathbb{F}_p$ within X.

CSIDH in one slide

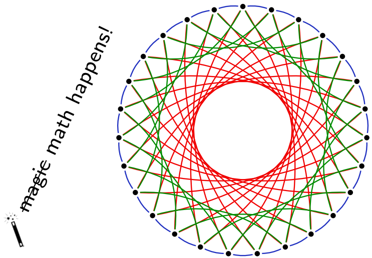
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p+1$ points$\}$.
- Look at the ℓ_i-isogenies defined over $\mathbb{F}_p$ within X.



$p = 419$
$\ell_1 = 3$
$\ell_2 = 5$
$\ell_3 = 7$

CSIDH in one slide

- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p+1$ points$\}$.
- Look at the ℓ_i-isogenies defined over $\mathbb{F}_p$ within X.

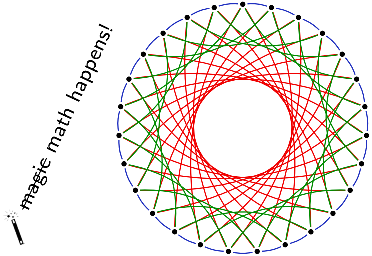


$p = 419$
$\ell_1 = 3$
$\ell_2 = 5$
$\ell_3 = 7$

- Walking "left" and "right" on any ℓ_i-subgraph is efficient.

CSIDH in one slide

- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p+1$ points$\}$.
- Look at the ℓ_i-isogenies defined over $\mathbb{F}_p$ within X.



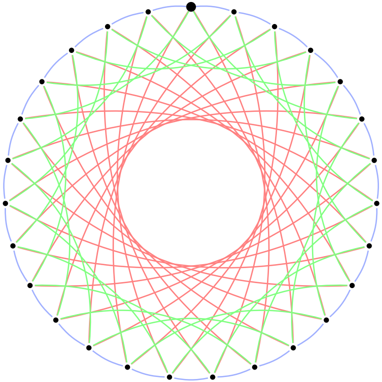
$p = 419$
$\ell_1 = 3$
$\ell_2 = 5$
$\ell_3 = 7$

- Walking "left" and "right" on any ℓ_i-subgraph is efficient.
- We can represent $E \in X$ as a single coefficient $A \in \mathbb{F}_p$.

CSIDH key exchange

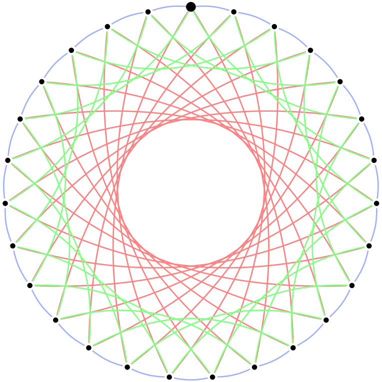
Alice

[+, +, −, −]



Bob

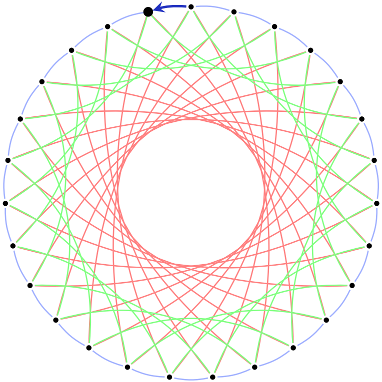
[−, +, −, −]



CSIDH key exchange

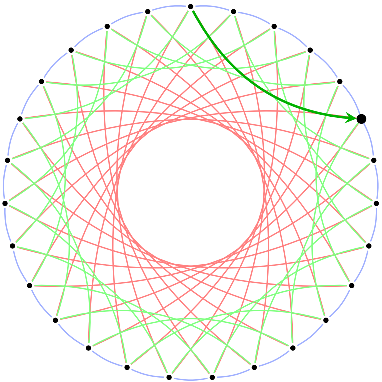
Alice

[+, +, −, −]
↑



Bob

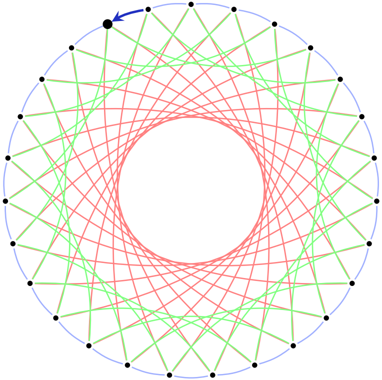
[−, +, −, −]
↑



CSIDH key exchange

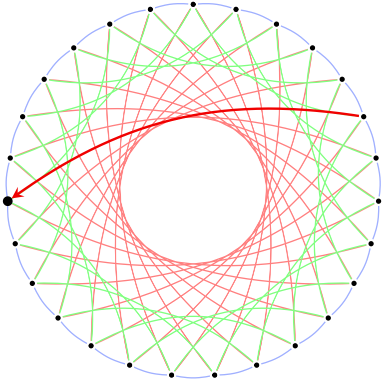
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



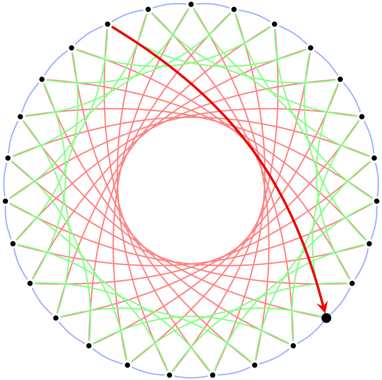
CSIDH key exchange

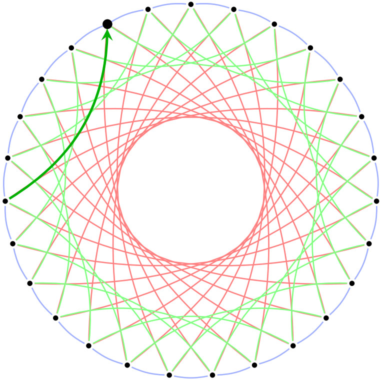
Alice

[+, +, −, −]
↑

Bob

[−, +, −, −]
↑

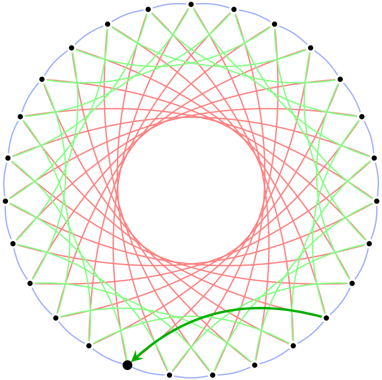




CSIDH key exchange

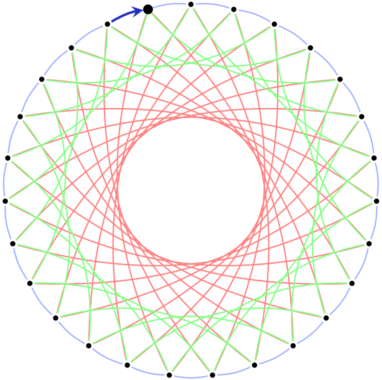
Alice

[+, +, −, −]

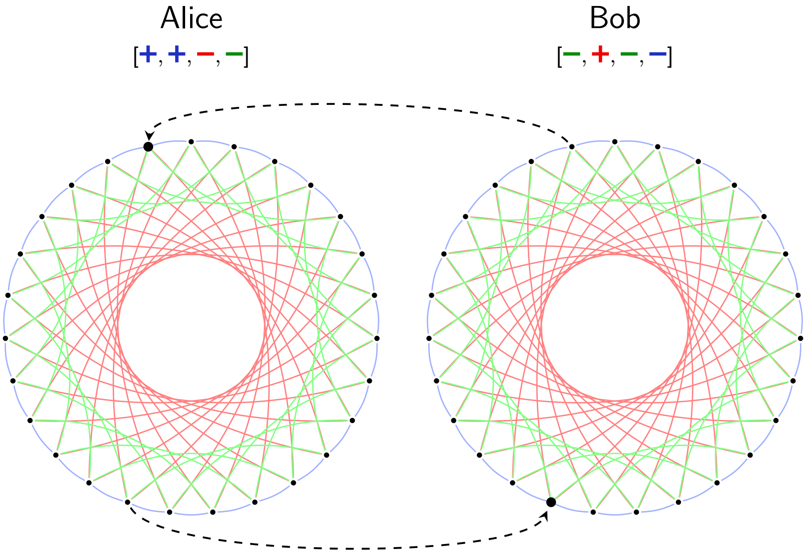


Bob

[−, +, −, −]



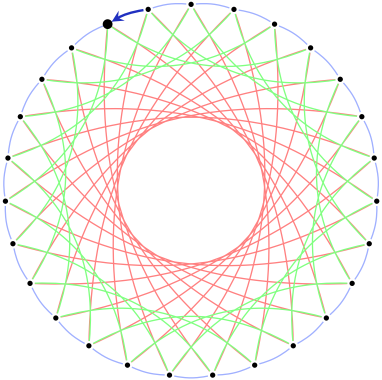
CSIDH key exchange



CSIDH key exchange

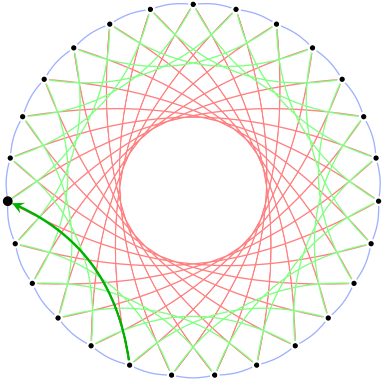
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



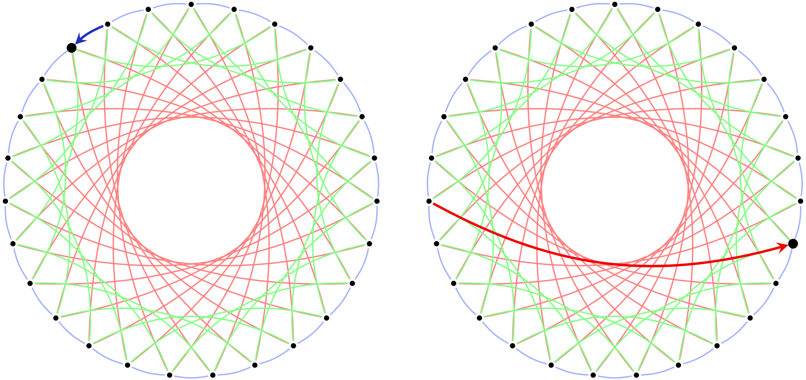
CSIDH key exchange

Alice

[+, +, −, −]

Bob

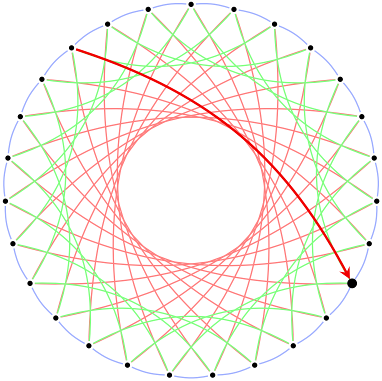
[−, +, −, −]



CSIDH key exchange

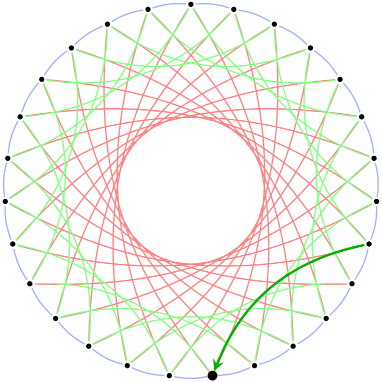
Alice

[+, +, −, −]
↑



Bob

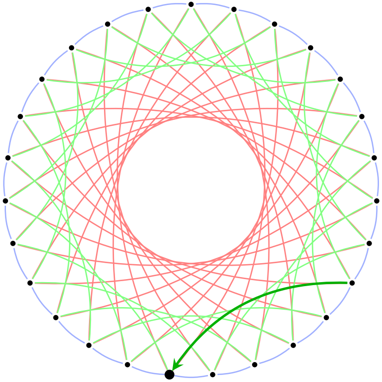
[−, +, −, −]
↑



CSIDH key exchange

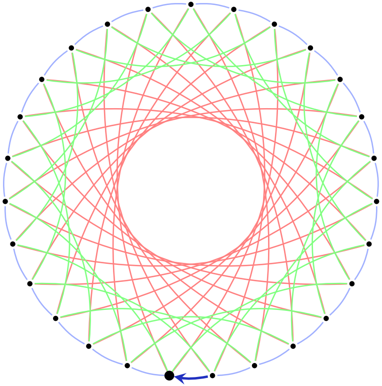
Alice

[+, +, −, −]
↑



Bob

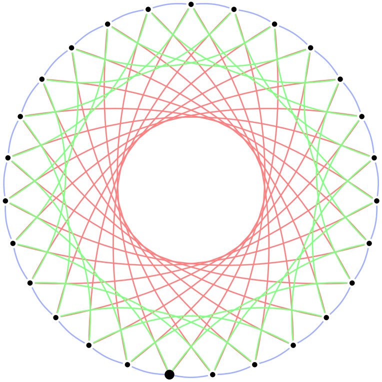
[−, +, −, −]
↑



CSIDH key exchange

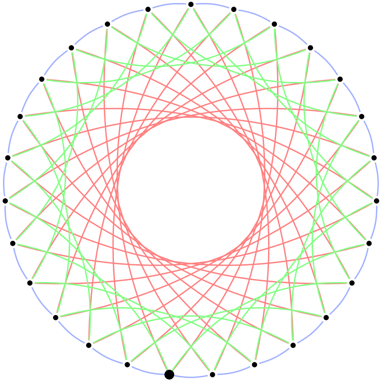
Alice

[+, +, −, −]



Bob

[−, +, −, −]



CSIDH action is commutative

Cycles are *compatible*: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of *total* step *counts* for each ℓ_i.

CSIDH action is commutative (hence the C)

Cycles are *compatible*: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of *total* step *counts* for each ℓ_i.

Example: [**+**, **+**, **−**, **−**, **−**, **+**, **−**, **−**] just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.
CSIDH private keys are vectors $(e_1, e_2, \ldots, e_n) \in [-m, m]^n$ for some m.

CSIDH action is commutative (hence the C)

Cycles are *compatible*: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of *total* step *counts* for each ℓ_i.

Example: $[+, +, -, -, -, +, -, -]$ just becomes $(+1, \quad 0, -3) \in \mathbb{Z}^3$.
CSIDH private keys are vectors $(e_1, e_2, \ldots, e_n) \in [-m, m]^n$ for some m.

Many paths are "useless".
Fun fact: Quotienting out trivial actions yields the *ideal-class group* $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

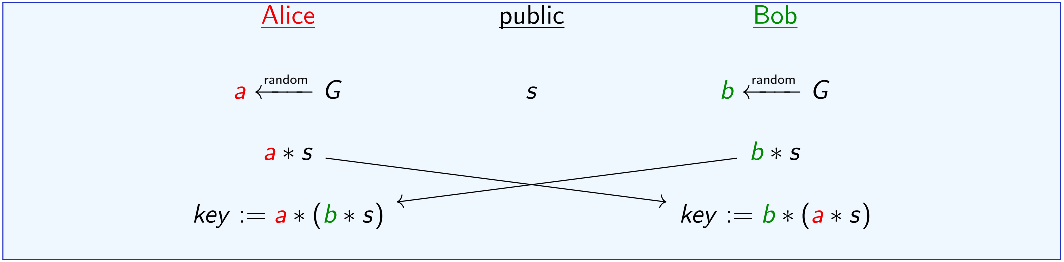
CSIDH action is commutative (hence the C)

Cycles are *compatible*: [right then left] = [left then right]
$\rightsquigarrow$ only need to keep track of *total* step *counts* for each ℓ_i.

Example: [+, +, −, −, −, +, −, −] just becomes $(+1, \ 0, -3) \in \mathbb{Z}^3$.
CSIDH private keys are vectors $(e_1, e_2, \ldots, e_n) \in [-m, m]^n$ for some m.

Many paths are "useless".
Fun fact: Quotienting out trivial actions yields the *ideal-class group* $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

There is a *group action* of $G = \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ on our *set of curves* X.



CSIDH security

Core problem:
Given $E, E' \in X$, *find* and compute *isogeny* $E \to E'$.

Size of key space:

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.
  (More precisely $\#\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ keys.)

Without quantum computer:

- Meet-in-the-middle variants: Time $O(\sqrt[4]{p})$.
  (2016 Delfs–Galbraith)

CSIDH security

Core problem:
Given $E, E' \in X$, *find* and compute *isogeny* $E \to E'$.

Size of key space:

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.
  (More precisely $\#\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ keys.)

Without quantum computer:

- Meet-in-the-middle variants: Time $\mathrm{O}(\sqrt[4]{p})$.
  (2016 Delfs–Galbraith)

With quantum computer:

- Abelian hidden-shift algorithms apply (2014 Childs–Jao–Soukharev)
  - These have subexponential complexity.
  - Not vulnerable to Shor's attack.

CSIDH security:

- Public-key validation:
  Quickly check that $E_A : y^2 = x^3 + Ax^2 + x$ has $p + 1$ points.

CSIDH-512 https://csidh.isogeny.org/

Definition:

- $p = 4\prod_{i=1}^{74} \ell_i - 1$ with $\ell_1, \ldots, \ell_{73}$ first 73 odd primes. $\ell_{74} = 587$.
- Exponents $-5 \leq e_i \leq 5$ for all $1 \leq i \leq 74$.

Sizes:

- Private keys: 32 bytes. (37 in current software for simplicity.)
- Public keys: 64 bytes (just one $\mathbb{F}_p$ element).

Performance on typical Intel Skylake laptop core:

- Clock cycles: about $12 \cdot 10^7$ per operation.
- ~~Somewhat more for constant-time implementations.~~
  https://ctidh.isogeny.org is fast and constant time.

Security:

- Pre-quantum: at least 128 bits.

CSIDH-512 https://csidh.isogeny.org/

Definition:

- $p = 4\prod_{i=1}^{74} \ell_i - 1$ with $\ell_1, \ldots, \ell_{73}$ first 73 odd primes. $\ell_{74} = 587$.
- Exponents $-5 \leq e_i \leq 5$ for all $1 \leq i \leq 74$.

Sizes:

- Private keys: 32 bytes. (37 in current software for simplicity.)
- Public keys: 64 bytes (just one $\mathbb{F}_p$ element).

Performance on typical Intel Skylake laptop core:

- Clock cycles: about $12 \cdot 10^7$ per operation.
- ~~Somewhat more for constant-time implementations.~~
  https://ctidh.isogeny.org is fast and constant time.

Security:

- Pre-quantum: at least 128 bits.
- Post-quantum: Several papers analyzing quantum approaches.
  (2018 Biasse–Iezzi-Jacobson, 2018-2020 Bonnetain–Schrottenloher, 2020 Peikert)
  All known attacks cost $\exp\left((\log p)^{1/2+o(1)}\right)$, improvements to sieving target the $o(1)$.
  Algorithms use "oracle calls". See https://quantum.isogeny.org for costs analysis.

Quadratic twists

E'/k is a *twist* of elliptic curve E/k if E' is isomorphic to E over $\bar{k}$.

For $E : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p \equiv 3 \bmod 4$ $E' : -y^2 = x^3 + Ax^2 + x$ is isomorphic to E via

$$(x, y) \mapsto (x, \sqrt{-1}y).$$

This map is defined over $\mathbb{F}_{p^2}$, so this is a *quadratic twist*.

Quadratic twists

E'/k is a *twist* of elliptic curve E/k if E' is isomorphic to E over $\bar{k}$.

For $E : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p \equiv 3 \bmod 4$ $E' : -y^2 = x^3 + Ax^2 + x$ is isomorphic to E via

$$(x, y) \mapsto (x, \sqrt{-1}y).$$

This map is defined over $\mathbb{F}_{p^2}$, so this is a *quadratic twist*.

E' is not in Weierstrass form (does not have the right shape).
E' is isomorphic to $E'' : y^2 = x^3 - Ax^2 + x$ via $(x, y) \mapsto (-x, y)$ over $\mathbb{F}_p$.

Quadratic twists

E'/k is a *twist* of elliptic curve E/k if E' is isomorphic to E over $\bar{k}$.

For $E : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p \equiv 3 \bmod 4$ $E' : -y^2 = x^3 + Ax^2 + x$ is isomorphic to E via

$$(x, y) \mapsto (x, \sqrt{-1}y).$$

This map is defined over $\mathbb{F}_{p^2}$, so this is a *quadratic twist*.

E' is not in Weierstrass form (does not have the right shape).
E' is isomorphic to $E'' : y^2 = x^3 - Ax^2 + x$ via $(x, y) \mapsto (-x, y)$ over $\mathbb{F}_p$.

Each $x \in \mathbb{F}_p$ satisfies one of

- $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$, thus there are two points $(x, \pm\sqrt{x^3 + Ax^2 + x})$ in $E(\mathbb{F}_p)$.
- $x^3 + Ax^2 + x$ is not a square in $\mathbb{F}_p$, thus there are two points $(x, \pm\sqrt{-(x^3 + Ax^2 + x)})$ in $E'(\mathbb{F}_p)$.
- $x^3 + Ax^2 + x = 0$, thus $(x, 0)$ is a point in $E(\mathbb{F}_p)$ and in $E'(\mathbb{F}_p)$.

Quadratic twists

E'/k is a *twist* of elliptic curve E/k if E' is isomorphic to E over $\bar{k}$.

For $E : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$ with $p \equiv 3 \bmod 4$ $E' : -y^2 = x^3 + Ax^2 + x$ is isomorphic to E via

$$(x, y) \mapsto (x, \sqrt{-1}y).$$

This map is defined over $\mathbb{F}_{p^2}$, so this is a *quadratic twist*.

E' is not in Weierstrass form (does not have the right shape).
E' is isomorphic to $E'' : y^2 = x^3 - Ax^2 + x$ via $(x, y) \mapsto (-x, y)$ over $\mathbb{F}_p$.

Each $x \in \mathbb{F}_p$ satisfies one of

- $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$, thus there are two points $(x, \pm\sqrt{x^3 + Ax^2 + x})$ in $E(\mathbb{F}_p)$.
- $x^3 + Ax^2 + x$ is not a square in $\mathbb{F}_p$, thus there are two points $(x, \pm\sqrt{-(x^3 + Ax^2 + x)})$ in $E'(\mathbb{F}_p)$.
- $x^3 + Ax^2 + x = 0$, thus $(x, 0)$ is a point in $E(\mathbb{F}_p)$ and in $E'(\mathbb{F}_p)$.

$\#E(\mathbb{F}_p) + \#E'(\mathbb{F}_p) = 2p + 2$, thus
$\#E(\mathbb{F}_p) = p + 1 - t$ implies $\#E'(\mathbb{F}_p) = p + 1 + t$.

Walking in the CSIDH graph

Taking a *"positive" step* on the ℓ_i-*subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p+1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Walking in the CSIDH graph

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Walking in the CSIDH graph

Taking a *"positive" step* on the ℓ_i*-subgraph*.

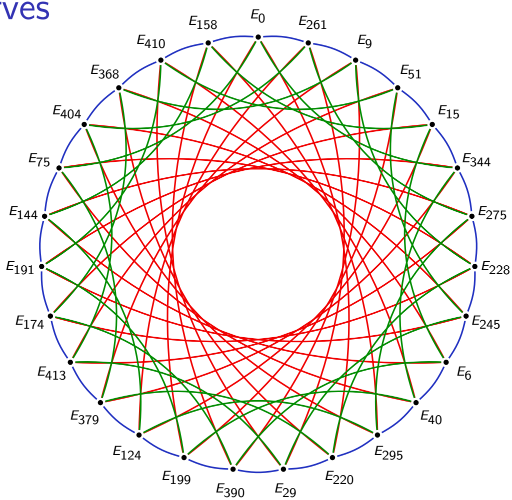
1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu's formulas.

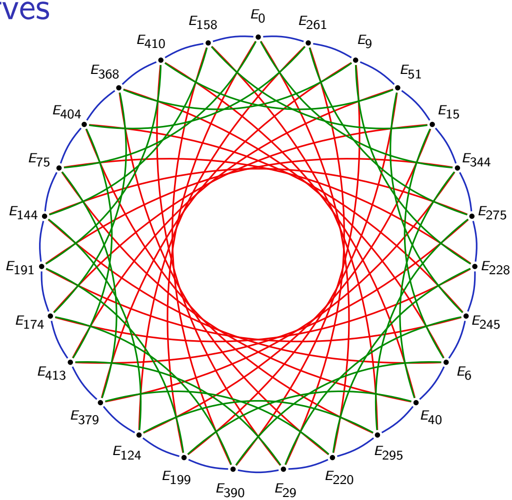
Upshot: With "x-only' arithmetic" everything happens *over* $\mathbb{F}_p$.
$\implies$ *Efficient* to implement! There are several more speedups, such as pushing points through isogenies.

Graphs of elliptic curves



Nodes: Supersingular elliptic curves $E_A\colon y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.

Graphs of elliptic curves



Nodes: Supersingular elliptic curves $E_A\colon y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Each E_A on the left has E_{-A} on the right.
Negative direction means: flip to twist, go positive direction, flip back.

Vélu's formulas

Let P have odd prime order ℓ on E_A.
For $1 \leq i < \ell$ let x_i be the x-coordinate of iP.
Let

$$\tau = \prod_{i=1}^{\ell-1} x_i, \quad \sigma = \sum_{i=1}^{\ell-1} \left(x_i - \frac{1}{x_i} \right), \quad f(x) = x \prod_{i=1}^{\ell-1} \frac{x x_i - 1}{x - x_i}.$$

Then the ℓ-isogeny with kernel $\langle P \rangle$ is given by

$$\varphi_\ell : E_A \to E_B, (x, y) \mapsto (f(x), c_0 y f'(x))$$

where $B = \tau(A - 3\sigma)$, and $c_0^2 = \tau$.

Main operation is to compute the x_i, just some elliptic-curve additions.
Note that $(\ell - i)P = -iP$ and both have the same x-coordinate.

Implementations often use *projective* formulas to avoid (or delay) inversions.

Montgomery curves have efficient arithmetic using only x-coordinates.

Disorientation faults in CSIDH

Gustavo Banegas, Juliane Krämer, Tanja Lange, Michael Meyer, Lorenz Panny,
Krijn Reijnders, Jana Sotáková, and Monika Trimoska

https://eprint.iacr.org/2022/1202

Steps in CSIDH computation

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu's formulas.

Steps in CSIDH computation

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

To find this point, we pick a random $x \in \mathbb{F}_p$, compute $z = x^3 + Ax^2 + x$ and check whether z is a square or not.
If it has the desired sign, multiply by $(p + 1)/\ell_i$ to (hopefully) get a point of order ℓ_i
– or repeat with new x.

Steps in CSIDH computation

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

To find this point, we pick a random $x \in \mathbb{F}_p$, compute $z = x^3 + Ax^2 + x$ and check whether z is a square or not.
If it has the desired sign, multiply by $(p + 1)/\ell_i$ to (hopefully) get a point of order ℓ_i
– or repeat with new x.

Implementations amortize this cost over multiple ℓ_i of the same orientation (sign).

Steps in CSIDH computation

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Taking a *"negative" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
   Same test as above to find such a point.
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

To find this point, we pick a random $x \in \mathbb{F}_p$, compute $z = x^3 + Ax^2 + x$ and check whether z is a square or not.
If it has the desired sign, multiply by $(p + 1)/\ell_i$ to (hopefully) get a point of order ℓ_i
– or repeat with new x.

Implementations amortize this cost over multiple ℓ_i of the same orientation (sign).
Knowing how often we take ℓ_i and in which orientation means knowing the key.

Computations in CSIDH

Require: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Ensure: $B \in \mathbb{F}_p$ such that $\prod [\mathfrak{l}_i]^{e_i} * E_A = E_B$

while some $e_i \neq 0$ **do**
    Sample a random $x \in \mathbb{F}_p$, defining a point P.
    Set $s \leftarrow \texttt{IsSquare}(x^3 + Ax^2 + x)$.
    Let $S = \{i \mid e_i \neq 0,\ \texttt{sign}(e_i) = s\}$. **Restart** with new x if S is empty.
    Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [\frac{p+1}{k}]P$.
    for each $i \in S$ **do**
        Set $k \leftarrow k/\ell_i$ and compute $R \leftarrow [k]Q$. If $R = \infty$, **skip** this i.
        Compute $\phi : E_A \to E_B$ with kernel $\langle R \rangle$.
        Set $A \leftarrow B$, $Q \leftarrow \phi(Q)$, and $e_i \leftarrow e_i - s$.
return A.

Computations in CSIDH the presence of attackers

Require: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Ensure: $B \in \mathbb{F}_p$ such that $\prod [\mathfrak{l}_i]^{e_i} * E_A = E_B$

while some $e_i \neq 0$ **do**
  Sample a random $x \in \mathbb{F}_p$, defining a point P.
  Set $s \leftarrow \texttt{IsSquare}(x^3 + Ax^2 + x)$.
  Let $S = \{i \mid e_i \neq 0,\ \texttt{sign}(e_i) = s\}$. **Restart** with new x if S is empty.
  Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [\frac{p+1}{k}]P$.
  for each $i \in S$ **do**
    Set $k \leftarrow k/\ell_i$ and compute $R \leftarrow [k]Q$. If $R = \infty$, **skip** this i.
    Compute $\phi : E_A \to E_B$ with kernel $\langle R \rangle$.
    Set $A \leftarrow B$, $Q \leftarrow \phi(Q)$, and $e_i \leftarrow e_i - s$.

return A.

An attacker can disturb the computation of $x^3 + Ax^2 + x$ or the $\texttt{IsSquare}$ test and disorient a whole batch of steps.

Computations in CSIDH the presence of attackers

Require: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Ensure: $B \in \mathbb{F}_p$ such that $\prod [\mathfrak{l}_i]^{e_i} * E_A = E_B$

- **while** some $e_i \neq 0$ **do**
  - Sample a random $x \in \mathbb{F}_p$, defining a point P.
  - Set $s \leftarrow \texttt{IsSquare}(x^3 + Ax^2 + x)$.
  - Let $S = \{i \mid e_i \neq 0,\ \texttt{sign}(e_i) = s\}$. **Restart** with new x if S is empty.
  - Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [\frac{p+1}{k}]P$.
  - **for each** $i \in S$ **do**
    - Set $k \leftarrow k/\ell_i$ and compute $R \leftarrow [k]Q$. If $R = \infty$, **skip** this i.
    - Compute $\phi : E_A \to E_B$ with kernel $\langle R \rangle$.
    - Set $A \leftarrow B$, $Q \leftarrow \phi(Q)$, and $e_i \leftarrow e_i - s$.
- **return** A.

An attacker can disturb the computation of $x^3 + Ax^2 + x$ or the IsSquare test and disorient a whole batch of steps.
Resulting curve E_t is **close** to E_B.

Computations in CSIDH the presence of attackers

Require: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Ensure: $B \in \mathbb{F}_p$ such that $\prod [\mathfrak{l}_i]^{e_i} * E_A = E_B$

while some $e_i \neq 0$ **do**
  Sample a random $x \in \mathbb{F}_p$, defining a point P.
  Set $s \leftarrow \texttt{IsSquare}(x^3 + Ax^2 + x)$.
  Let $S = \{i \mid e_i \neq 0,\ \texttt{sign}(e_i) = s\}$. **Restart** with new x if S is empty.
  Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [\frac{p+1}{k}]P$.
  for each $i \in S$ **do**
    Set $k \leftarrow k/\ell_i$ and compute $R \leftarrow [k]Q$. If $R = \infty$, **skip** this i.
    Compute $\phi : E_A \rightarrow E_B$ with kernel $\langle R \rangle$.
    Set $A \leftarrow B$, $Q \leftarrow \phi(Q)$, and $e_i \leftarrow e_i - s$.
return A.

An attacker can disturb the computation of $x^3 + Ax^2 + x$ or the IsSquare test and disorient a whole batch of steps.
Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

How to recover key from faulty curves?

Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

To get to E_B we need to compute these in positive direction if we flipped s from $-$ to $+$ (we walked in the negative direction for primes that wanted to walk in the positive direction).

Likewise, we need to compute them in the negative direction if we flipped from $+$ to $-$.

Finding S tells us the signs of those e_i.

How to recover key from faulty curves?

Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

To get to E_B we need to compute these in positive direction if we flipped s from $-$ to $+$ (we walked in the negative direction for primes that wanted to walk in the positive direction).

Likewise, we need to compute them in the negative direction if we flipped from $+$ to $-$.

Finding S tells us the signs of those e_i.

We can fault in different rounds to get all e_i – first round faults get $|e_i| \geq 1$, second round faults get $|e_i| \geq 2, \ldots$

How to recover key from faulty curves?

Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

To get to E_B we need to compute these in positive direction if we flipped s from $-$ to $+$ (we walked in the negative direction for primes that wanted to walk in the positive direction).

Likewise, we need to compute them in the negative direction if we flipped from $+$ to $-$.

Finding S tells us the signs of those e_i.

We can fault in different rounds to get all e_i – first round faults get $|e_i| \geq 1$, second round faults get $|e_i| \geq 2$, ... but we don't actually know how many $+$ and how many $-$ we have done when we fault in round > 1.

How to recover key from faulty curves?

Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

To get to E_B we need to compute these in positive direction if we flipped s from $-$ to $+$ (we walked in the negative direction for primes that wanted to walk in the positive direction).

Likewise, we need to compute them in the negative direction if we flipped from $+$ to $-$.

Finding S tells us the signs of those e_i.

We can fault in different rounds to get all e_i – first round faults get $|e_i| \geq 1$, second round faults get $|e_i| \geq 2$, ... but we don't actually know how many $+$ and how many $-$ we have done when we fault in round > 1.
In round 2 we might have done $++$, $+-$, $-+$, or $--$.

How to recover key from faulty curves?

Resulting curve E_t is **close** to E_B.
Off by exactly $2\ell_i$ isogenies for $i \in S$ when the fault happened.

To get to E_B we need to compute these in positive direction if we flipped s from $-$ to $+$ (we walked in the negative direction for primes that wanted to walk in the positive direction).

Likewise, we need to compute them in the negative direction if we flipped from $+$ to $-$.

Finding S tells us the signs of those e_i.

We can fault in different rounds to get all e_i – first round faults get $|e_i| \geq 1$, second round faults get $|e_i| \geq 2$, ... but we don't actually know how many $+$ and how many $-$ we have done when we fault in round > 1.
In round 2 we might have done $++$, $+-$, $-+$, or $--$.
Middle 2 options give curves we have seen as results in round 1.

Let $E^{i,+}$ and $E^{i,-}$ denote the curves when faulting the i-th occurrence of $+$ and $-$, respectively.

Cost of this attack

At least one of the faulty curves in round 1 has no more than $n/2$ elements in S. Brute force search takes

$$\binom{n}{n/2.}$$

For CSIDH-512 $n = 74$, so $\binom{74}{37} \equiv 2^{70}$.

Cost of this attack

At least one of the faulty curves in round 1 has no more than $n/2$ elements in S. Brute force search takes

$$\binom{n}{n/2.}$$

For CSIDH-512 $n = 74$, so $\binom{74}{37} \equiv 2^{70}$.

We can walk from E_B and E_t and meet in the middle for

$$\sqrt{\binom{n}{n/2}}.$$

Cost of this attack

At least one of the faulty curves in round 1 has no more than $n/2$ elements in S. Brute force search takes

$$\binom{n}{n/2.}$$

For CSIDH-512 $n = 74$, so $\binom{74}{37} \equiv 2^{70}$.

We can walk from E_B and E_t and meet in the middle for

$$\sqrt{\binom{n}{n/2}}.$$

But there is a lot more information we can get!

Cost of this attack

At least one of the faulty curves in round 1 has no more than $n/2$ elements in S. Brute force search takes

$$\binom{n}{n/2.}$$

For CSIDH-512 $n = 74$, so $\binom{74}{37} \equiv 2^{70}$.

We can walk from E_B and E_t and meet in the middle for

$$\sqrt{\binom{n}{n/2}}.$$

But there is a lot more information we can get!

$E^{1,+}$ and $E^{2,+}$ differ by those ℓ_i that have exactly $e_i = 1$.
$E^{2,+}$ and $E^{2,+}$ differ by those ℓ_i that have exactly $e_i = 2$.
$\vdots$
These gaps are much smaller, on average $n/(2m+1)$.

Even more information

Taking a *“positive” step* on the ℓ_i-*subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu’s formulas.

Even more information

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu's formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but

Even more information

Taking a *"positive" step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p+1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but
with probability $1/3$ we miss $\ell_1 = 3$ in the order of the point,

Even more information

Taking a *"positive" step* on the ℓ_i-*subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p+1$, so $[(p+1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but
with probability $1/3$ we miss $\ell_1 = 3$ in the order of the point,
with probability $1/5$ we miss $\ell_2 = 5$ in the order of the point, ...

Even more information

Taking a *“positive” step* on the ℓ_i*-subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle(x, y)\rangle$ using Vélu’s formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but
with probability $1/3$ we miss $\ell_1 = 3$ in the order of the point,
with probability $1/5$ we miss $\ell_2 = 5$ in the order of the point, . . .
We get clouds of curves at distance 1 or 2 primes from $E^{1,+}$ and E^{1-}.
These very efficiently reveal orientations of small primes and thus reduce the search space.

Even more information

Taking a *"positive" step* on the ℓ_i-*subgraph*.

1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but
with probability $1/3$ we miss $\ell_1 = 3$ in the order of the point,
with probability $1/5$ we miss $\ell_2 = 5$ in the order of the point, ...
We get clouds of curves at distance 1 or 2 primes from $E^{1,+}$ and E^{1-}.
These very efficiently reveal orientations of small primes and thus reduce the search space.

Later rounds have the same, but also have some 'late comers' pointing in the wrong direction.
$e_i = 1$ will have $\ell_1 = 3$ appear near $E^{2,+}$ with probability $1/3$, when it was missed in round 1.

Even more information

Taking a *"positive" step* on the *ℓ_i-subgraph*.

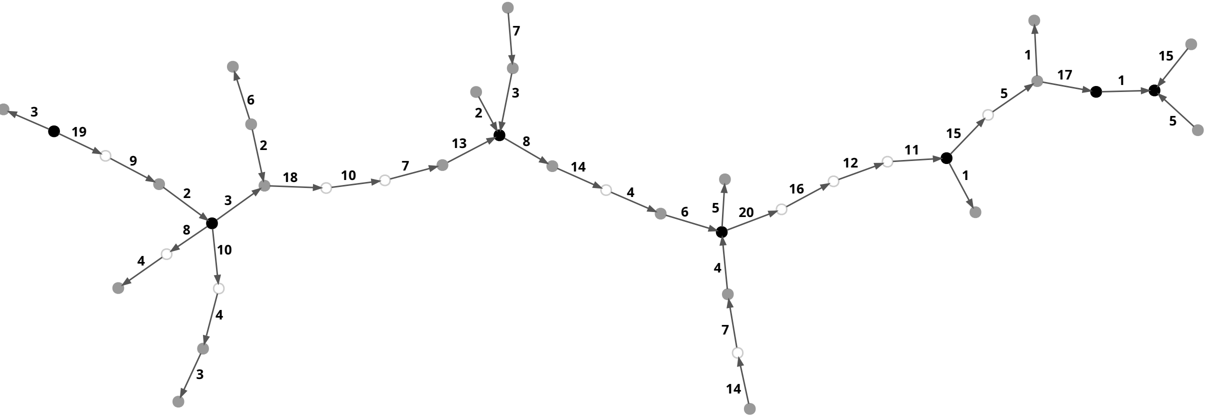
1. Find a point $(x, y) \in E$ of *order* ℓ_i with $x, y \in \mathbb{F}_p$.
   The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
   Sample a new point if you get ∞ (probability $1/\ell_i$).
2. Compute the *isogeny* with *kernel* $\langle (x, y) \rangle$ using Vélu's formulas.

Repeatedly faulting round 1 will give many copies of $E^{1,+}$ and $E^{1,-}$, but
with probability $1/3$ we miss $\ell_1 = 3$ in the order of the point,
with probability $1/5$ we miss $\ell_2 = 5$ in the order of the point, ...
We get clouds of curves at distance 1 or 2 primes from $E^{1,+}$ and E^{1-}.
These very efficiently reveal orientations of small primes and thus reduce the search space.

Later rounds have the same, but also have some 'late comers' pointing in the wrong direction.
$e_i = 1$ will have $\ell_1 = 3$ appear near $E^{2,+}$ with probability $1/3$, when it was missed in round 1.

Our tool, `pubcrawl`, does MitM searches in neighborhoods of curves.

Graph for toy CSIDH-103 ($n = 21, m = 3$)



Black: $E^{1,+}, E^{2,+}, E^{3,+}, E_B, E^{3,-}, E^{2,-}, E^{1,-}$;
gray: Other faulty curves in neighborhood;
white: intermediate curves found with `pubcrawl`.

See the paper for

- How to induce such faults.
  Note: this attack uses a lot of nice math but starts from a physical attack, so the attacker needs physical access.
- Other keyspaces incl. CTIDH.
- Probabilities and cost estimates.
- How to read off the key from `pubcrawl` and the graphs.
- What you can still do if you get only hash(E_t) instead of E_t.
- Speedups.

https://eprint.iacr.org/2022/1202.

CSIDH with countermeasures

Require: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Ensure: $B \in \mathbb{F}_p$ such that $\prod\, [\mathfrak{l}_i]^{e_i} * E_A = E_B$

while some $e_i \neq 0$ **do**
    Sample a random $x \in \mathbb{F}_p$, defining a point P.
    Set $z \leftarrow x^3 + Ax^2 + x$, $\tilde{y} \leftarrow z^{(p+1)/4}$.
    Set $s \leftarrow 1$ if $\tilde{y}^2 = z$, $s \leftarrow -1$ if $\tilde{y}^2 = -z$, $s \leftarrow 0$ otherwise.
    Let $S = \{i \mid e_i \neq 0,\ \mathrm{sign}(e_i) = s\}$. **Restart** with new x if S is empty.
    Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q' = (X_{Q'} : Z_{Q'}) \leftarrow [\frac{p+1}{k}]P$.
    Compute $z' \leftarrow x^3 + Ax^2 + x$.
    Set $X_Q \leftarrow s \cdot z' \cdot X_{Q'}$, $Z_Q \leftarrow \tilde{y}^2 \cdot Z_{Q'}$.
    Set $Q = (X_Q : Z_Q)$.
    for each $i \in S$ **do**
        Set $k \leftarrow k/\ell_i$ and compute $R \leftarrow [k]Q$. If $R = \infty$, **skip** this i.
        Compute $\phi : E_A \to E_B$ with kernel $\langle R \rangle$.
        Set $A \leftarrow B$, $Q \leftarrow \phi(Q)$, and $e_i \leftarrow e_i - s$.

return A.

This uses z in computation rather than just s, faults make us move outside set of curves.

Further information

- YouTube channel Tanja Lange: Post-quantum cryptography.
- Isogeny-based cryptography school.
- https://2017.pqcrypto.org/school: PQCRYPTO summer school with 21 lectures on video, slides, and exercises.
- https://2017.pqcrypto.org/exec and https://pqcschool.org/index.html: Executive school (less math, more perspective).
- https://pqcrypto.org our overview page.
- ENISA report on PQC, co-authored.
- https://pqcrypto.eu.org: PQCRYPTO EU Project.
  - PQCRYPTO recommendations.
  - Free software libraries (libpqcrypto, pqm4, pqhw).
  - Many reports, scientific articles, (overview) talks.
- Quantum Threat Timeline from Global Risk Institute, 2019; 2021 update.
- Status of quantum computer development (by German BSI).
- NIST PQC competition.
- PQCrypto 2016, PQCrypto 2017, PQCrypto 2018, PQCrypto 2019, PQCrypto 2020, PQCrypto 2021, PQCrypto 2022 with many slides and videos online.