Isogeny-based Cryptography

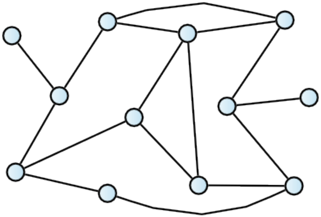
Lorenz Panny

Technische Universität München

QSI Spring School, Porto, 15 March 2024

Big picture

- Isogenies are a source of exponentially-sized graphs.



Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.

Big picture

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

Big picture

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

Finding graphs with *almost* all of these properties is easy —
but getting all at once seems rare.

Crypto on graphs?

Diffie–Hellman key exchange 1976

Public parameters:

- a finite group G (traditionally $\mathbb{F}_p^*$, today elliptic curves)
- an element $g \in G$ of prime order q

Diffie–Hellman key exchange 1976

Public parameters:

- a finite group G (traditionally $\mathbb{F}_p^*$, today elliptic curves)
- an element $g \in G$ of prime order q

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \dots q-1\}$		$b \xleftarrow{\text{random}} \{0 \dots q-1\}$
g^a		g^b
$s := (g^b)^a$		$s := (g^a)^b$

Diffie–Hellman key exchange 1976

Public parameters:

- a finite group G (traditionally $\mathbb{F}_p^*$, today elliptic curves)
- an element $g \in G$ of prime order q

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \dots q-1\}$		$b \xleftarrow{\text{random}} \{0 \dots q-1\}$
g^a		g^b
$s := (g^b)^a$		$s := (g^a)^b$

Fundamental reason this works: $\cdot^a$ and $\cdot^b$ are commutative!

Diffie–Hellman: Bob vs. Eve

Bob

1. Set $t \leftarrow g$.
2. Set $t \leftarrow t \cdot g$.
3. Set $t \leftarrow t \cdot g$.
4. Set $t \leftarrow t \cdot g$.

...

$b-2$. Set $t \leftarrow t \cdot g$.

$b-1$. Set $t \leftarrow t \cdot g$.

b. Publish $B \leftarrow t \cdot g$.

Diffie–Hellman: Bob vs. Eve

Bob

1. Set $t \leftarrow g$.
2. Set $t \leftarrow t \cdot g$.
3. Set $t \leftarrow t \cdot g$.
4. Set $t \leftarrow t \cdot g$.

...

$b-2$. Set $t \leftarrow t \cdot g$.

$b-1$. Set $t \leftarrow t \cdot g$.

b. Publish $B \leftarrow t \cdot g$.

Is this a good idea?

Diffie–Hellman: Bob vs. Eve

<u>Bob</u>

1. Set $t \leftarrow g$.
2. Set $t \leftarrow t \cdot g$.
3. Set $t \leftarrow t \cdot g$.
4. Set $t \leftarrow t \cdot g$.

...

$b{-}2$. Set $t \leftarrow t \cdot g$.

$b{-}1$. Set $t \leftarrow t \cdot g$.

b. Publish $B \leftarrow t \cdot g$.

<u>**Attacker** Eve</u>

1. Set $t \leftarrow g$. If $t = B$ return 1.
2. Set $t \leftarrow t \cdot g$. If $t = B$ return 2.
3. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.
4. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.

...

$b{-}2$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b{-}2$.

$b{-}1$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b{-}1$.

b. Set $t \leftarrow t \cdot g$. If $t = B$ return b.

$b{+}1$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b + 1$.

$b{+}2$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b + 2$.

...

Diffie–Hellman: Bob vs. Eve

Bob

1. Set $t \leftarrow g$.
2. Set $t \leftarrow t \cdot g$.
3. Set $t \leftarrow t \cdot g$.
4. Set $t \leftarrow t \cdot g$.

...

$b-2$. Set $t \leftarrow t \cdot g$.

$b-1$. Set $t \leftarrow t \cdot g$.

b. Publish $B \leftarrow t \cdot g$.

Attacker Eve

1. Set $t \leftarrow g$. If $t = B$ return 1.
2. Set $t \leftarrow t \cdot g$. If $t = B$ return 2.
3. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.
4. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.

...

$b-2$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b-2$.

$b-1$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b-1$.

b. Set $t \leftarrow t \cdot g$. If $t = B$ return b.

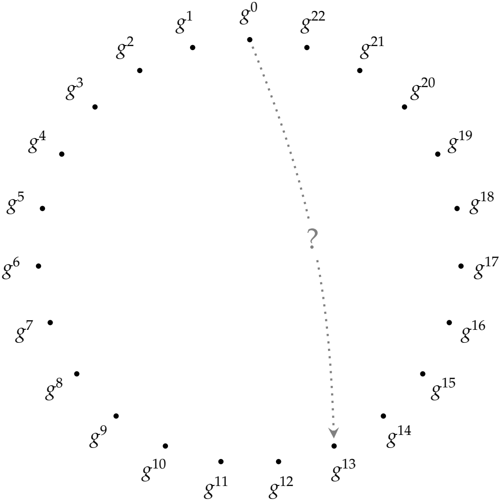
$b+1$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b+1$.

$b+2$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b+2$.

...

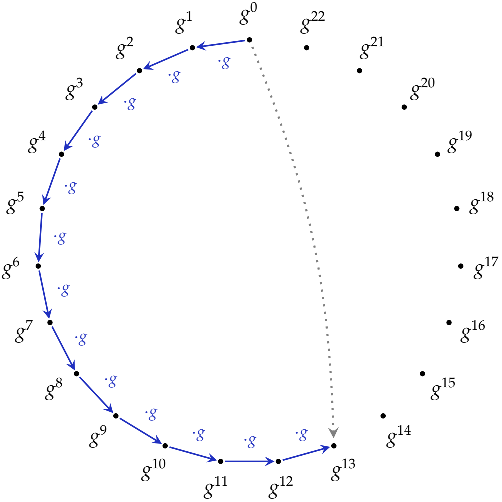
Effort for both: $O(\#G)$. Bob needs to be smarter.

(This attacker is also kind of dumb, but that doesn't matter for my point here.)



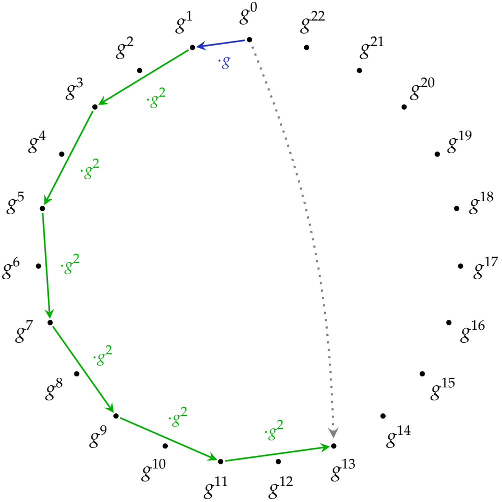
Bob computes his public key g^{13} from g.

multiply



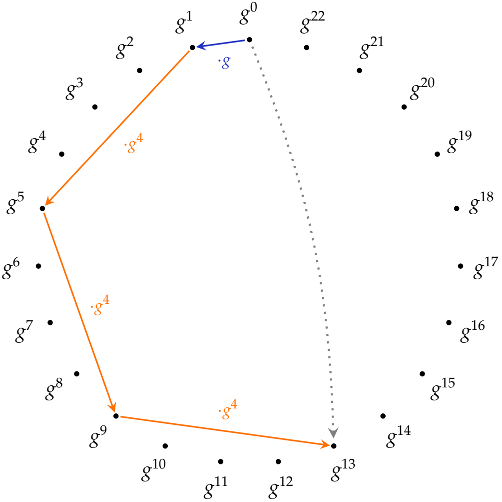
Bob computes his public key g^{13} from g.

Square-and-multiply



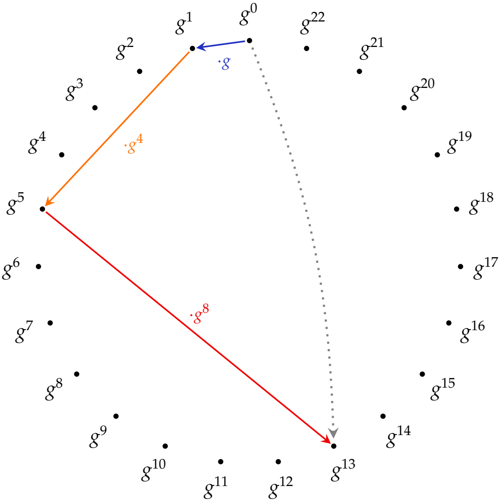
Bob computes his public key g^{13} from g.

Square-and-multiply-and-square-and-multiply



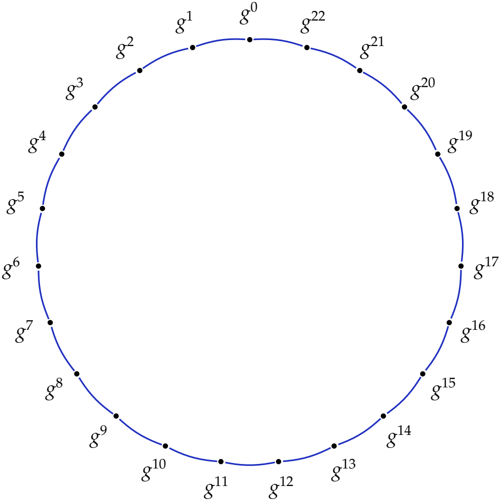
Bob computes his public key g^{13} from g.

Square-and-multiply-and-square-and-multiply-and-squ

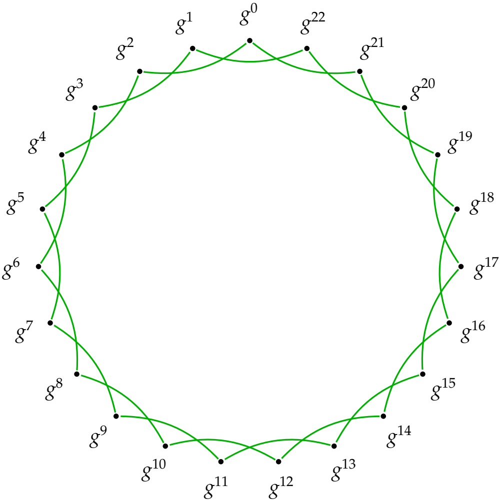


Bob computes his public key g^{13} from g.

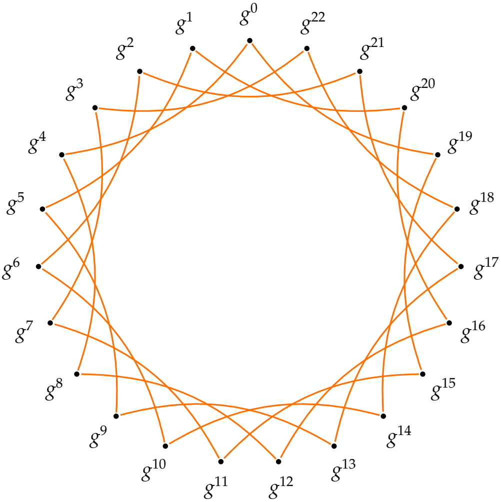
Square-and-multiply as graphs



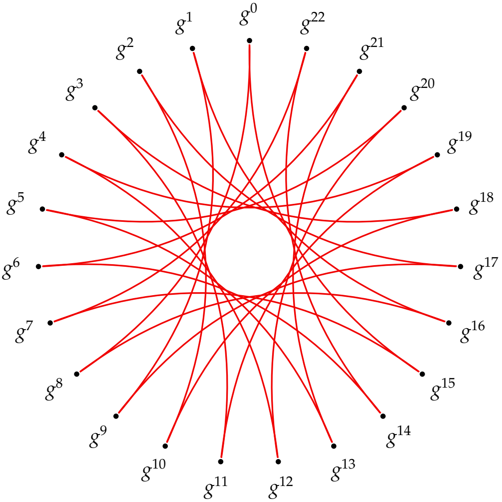
Square-and-multiply as graphs



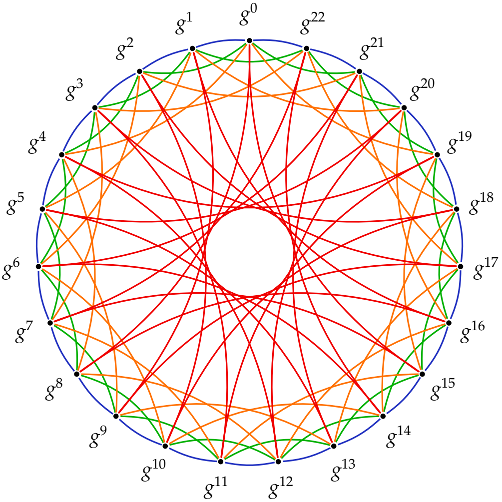
Square-and-multiply as graphs



Square-and-multiply as graphs



Square-and-multiply as a graph



Crypto on graphs?

We’ve been doing it all the time!

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

With square-and-multiply, computing $\alpha \mapsto g^{\alpha}$ takes $\Theta(\log \alpha)$.

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

With square-and-multiply, computing $\alpha \mapsto g^\alpha$ takes $\Theta(\log \alpha)$.

For well-chosen groups, computing $g^\alpha \mapsto \alpha$ takes $\Theta(\sqrt{\#G})$.

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

With square-and-multiply, computing $\alpha \mapsto g^\alpha$ takes $\Theta(\log \alpha)$.

For well-chosen groups, computing $g^\alpha \mapsto \alpha$ takes $\Theta(\sqrt{\#G})$.

$\rightsquigarrow$ Exponential separation!

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

With square-and-multiply, computing $\alpha \mapsto g^\alpha$ takes $\Theta(\log \alpha)$.

For well-chosen groups, computing $g^\alpha \mapsto \alpha$ takes $\Theta(\sqrt{\#G})$.

$\rightsquigarrow$ Exponential separation!

...and they lived happily ever after**?**

The fast mixing requirement

Fast mixing: paths of length $\log(\#\,\text{nodes})$ to everywhere.

With square-and-multiply, computing $\alpha \mapsto g^\alpha$ takes $\Theta(\log \alpha)$.

For well-chosen groups, computing $g^\alpha \mapsto \alpha$ takes $\Theta(\sqrt{\#G})$.

$\rightsquigarrow$ Exponential separation!

...and they lived happily ever after**?**

Shor's quantum algorithm computes α from g^α in any group in polynomial time.

In some cases,

<u>isogeny graphs</u>

can replace DLP-based constructions post-quantumly.

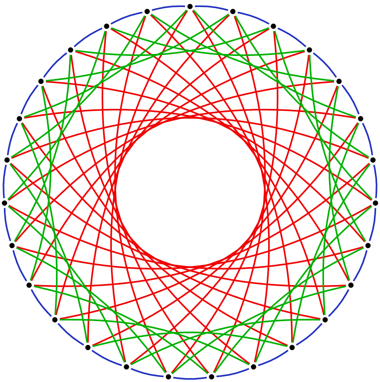
In some cases,

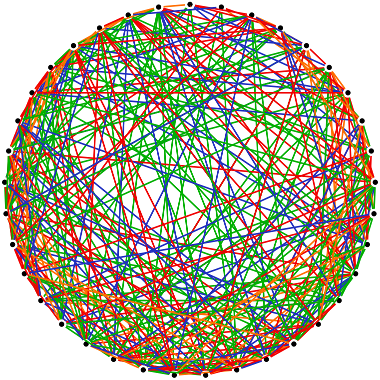
isogeny graphs

can replace $\underset{\textit{some}}{\curlywedge}$ DLP-based constructions post-quantumly.

The beauty and the beast

Components of particular isogeny graphs look like this:

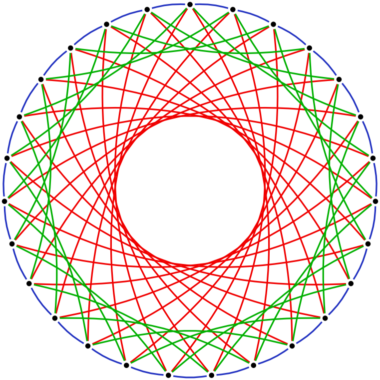


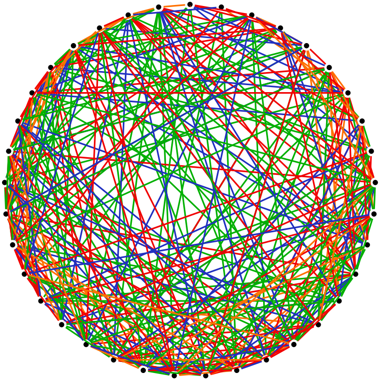


Which of these is good for crypto?

The beauty and the beast

Components of particular isogeny graphs look like this:





Which of these is good for crypto? **Both.** ⌣̈

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies.
- The CGL hash function.
- The CSIDH non-interactive key exchange.
- Hardness of isogeny problems, and reductions.
- The SQIsign signature scheme.
- Transcending to higher dimensions.

Stand back!

We're going to do math.

Elliptic curves

An elliptic curve over a field F of characteristic $\notin \{2, 3\}$ is* an equation of the form

$$E\colon\ y^2 = x^3 + ax + b$$

with $a, b \in F$ such that $4a^3 + 27b^2 \neq 0$.

Elliptic curves

An elliptic curve over a field F of characteristic $\notin \{2, 3\}$ is* an equation of the form

$$E\colon\; y^2 = x^3 + ax + b$$

with $a, b \in F$ such that $4a^3 + 27b^2 \neq 0$.

A point on E is a solution (x, y), or the "fake" point ∞.

Elliptic curves

An elliptic curve over a field F of characteristic $\notin \{2, 3\}$ is* an equation of the form

$$E\colon\ y^2 = x^3 + ax + b$$

with $a, b \in F$ such that $4a^3 + 27b^2 \neq 0$.

A point on E is a solution (x, y), or the "fake" point ∞.

E is an abelian group: we can "add" points.

Elliptic curves

An elliptic curve over a field F of characteristic $\notin \{2,3\}$ is* an equation of the form

$$E\colon\ y^2 = x^3 + ax + b$$

with $a, b \in F$ such that $4a^3 + 27b^2 \neq 0$.

A point on E is a solution (x, y), or the "fake" point ∞.

E is an abelian group: we can "add" points.

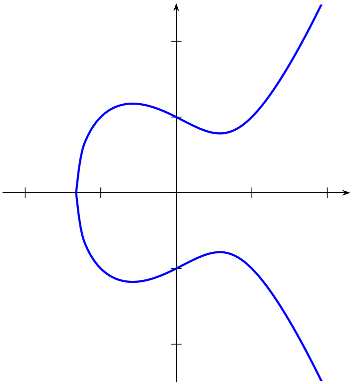
- The neutral element is ∞.
- The inverse of (x, y) is $(x, -y)$.
- The sum of (x_1, y_1) and (x_2, y_2) is

do **not** remember these formulas!

$$\left(\lambda^2 - x_1 - x_2,\ \lambda(2x_1 + x_2 - \lambda^2) - y_1\right)$$

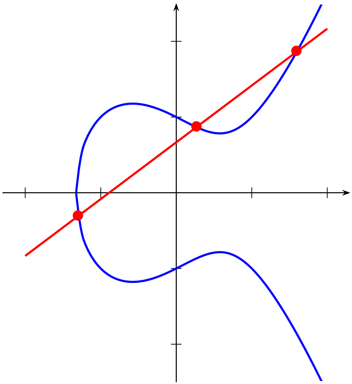
where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $x_1 \neq x_2$ and $\lambda = \frac{3x_1^2 + a}{2y_1}$ otherwise.

Elliptic curves (picture over $\mathbb{R}$)



The elliptic curve $y^2 = x^3 - x + 1$ over $\mathbb{R}$.

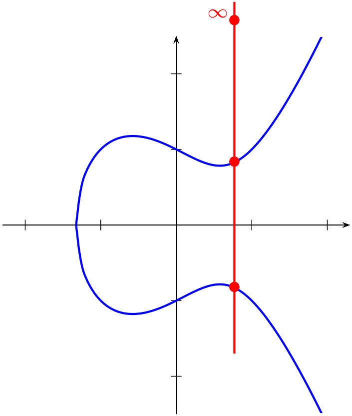
Elliptic curves (picture over $\mathbb{R}$)



Addition law:

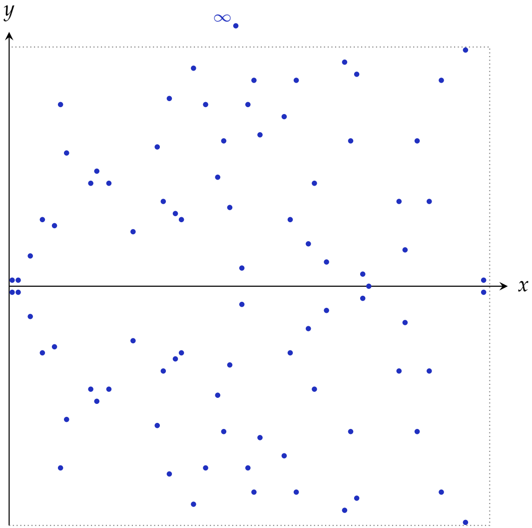
$$P + Q + R = \infty \iff \{P, Q, R\} \text{ on a straight line.}$$

Elliptic curves (picture over $\mathbb{R}$)



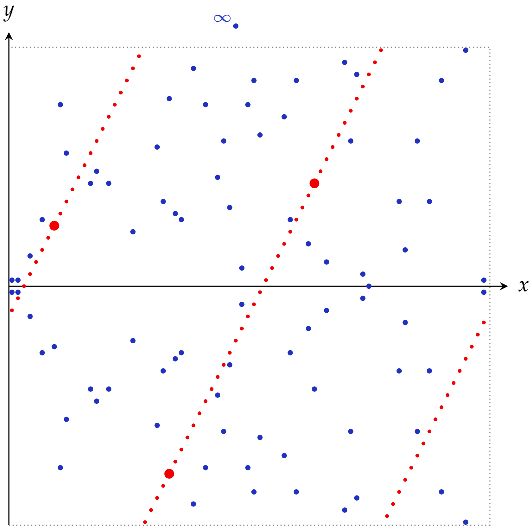
The *point at infinity* ∞ lies on every vertical line.

Elliptic curves (picture over $\mathbb{F}_p$)



The same curve $y^2 = x^3 - x + 1$ over the finite field $\mathbb{F}_{79}$.

Elliptic curves (picture over $\mathbb{F}_p$)



The addition law of $y^2 = x^3 - x + 1$ over the finite field $\mathbb{F}_{79}$.

In SageMath:

```
sage: E = EllipticCurve(GF(101), [5,6,7,8,9])
sage: E
Elliptic Curve defined by
            y^2 + 5*x*y + 7*y = x^3 + 6*x^2 + 8*x + 9
            over Finite Field of size 101
```

In SageMath:

```
sage: E = EllipticCurve(GF(101), [5,6,7,8,9])
sage: E
Elliptic Curve defined by
            y^2 + 5*x*y + 7*y = x^3 + 6*x^2 + 8*x + 9
            over Finite Field of size 101
sage: P = E(3, 18)       # constructing points
sage: Q = E(8, 75)
sage: P + Q              # point addition
(73 : 24 : 1)
```

In SageMath:

```
sage: E = EllipticCurve(GF(101), [5,6,7,8,9])
sage: E
Elliptic Curve defined by
            y^2 + 5*x*y + 7*y = x^3 + 6*x^2 + 8*x + 9
            over Finite Field of size 101
sage: P = E(3, 18)      # constructing points
sage: Q = E(8, 75)
sage: P + Q             # point addition
(73 : 24 : 1)
sage: P - P
(0 : 1 : 0)              # point at infinity
```

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

Define scalar multiplication $[n]P := \underbrace{P + \cdots + P}_{n \text{ times}}$. (Use double-and-add!)

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

Define scalar multiplication $[n]P := \underbrace{P + \cdots + P}_{n \text{ times}}$. (Use double-and-add!)

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \ldots \ell{-}1\}$		$b \xleftarrow{\text{random}} \{0 \ldots \ell{-}1\}$
$[a]P$		$[b]P$
$s := [a]([b]P)$	$\longleftarrow$ equal! $\longrightarrow$	$s := [b]([a]P)$

Fields of definition

Generally, things can be defined over extension fields:
For example, $(0, \sqrt{-1})$ is a point of $y^2 = x^3 - 1$.

Let k be a field.

An elliptic curve/point/isogeny is defined over k or k-rational if the coefficients in its equation/formula lie in k.

We write E/k for "E is defined over k".

Fields of definition

Generally, things can be defined over extension fields:
For example, $(0, \sqrt{-1})$ is a point of $y^2 = x^3 - 1$.

Let k be a field.

An elliptic curve/point/isogeny is defined over k or k-rational if the coefficients in its equation/formula lie in k.
We write E/k for "E is defined over k".

For E/k, write $E(k)$ for the set of points of E defined over k.

Note: Simply writing E means $E(\bar{k})$, i.e., points over *all* extension fields.

In SageMath:

Everything happens over the specified field of definition:

In SageMath:

Everything happens over the specified field of definition:

```
sage: E = EllipticCurve(GF(101), [0,5,0,1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + 5*x^2 + x
                over Finite Field of size 101
sage: F.<t> = GF(101^2)
sage: E(11, 69*t + 64)
ValueError: 69*t + 64 is not in the image of #...
sage: EE = E.change_ring(F)
sage: EE(11, 69*t + 64)
(11 : 69*t + 64 : 1)
```

Isogenies

Isogenies

...are just fancily-named

nice maps

between elliptic curves.

Isogenies

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

Isogenies

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

- given by rational functions.

Isogenies

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

- given by rational functions.
- a group homomorphism.

Isogenies

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

- given by rational functions.
- a group homomorphism.

Reminder:

A rational function is $f(x, y)/g(x, y)$ where f, g are polynomials.

A group homomorphism φ satisfies $\varphi(P + Q) = \varphi(P) + \varphi(Q)$.

Isogenies

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

- given by rational functions.
- a group homomorphism.

Reminder:

A rational function is $f(x, y)/g(x, y)$ where f, g are polynomials.

A group homomorphism φ satisfies $\varphi(P + Q) = \varphi(P) + \varphi(Q)$.

The kernel of an isogeny $\varphi\colon E \to E'$ is $\{P \in E \,:\, \varphi(P) = \infty\}$.
The degree of a separable* isogeny is the size of its kernel.

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Example #1: $(x, y) \mapsto \left(\frac{x^3-4x^2+30x-12}{(x-2)^2}, \frac{x^3-6x^2-14x+35}{(x-2)^3} \cdot y \right)$
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 $\{(2, 9), (2, -9), \infty\}$.

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

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 (examples)

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

- given by rational functions.
- a group homomorphism.

Example #3: For each $m \neq 0$, the multiplication-by-m map

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

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Example #3: For each $m \neq 0$, the multiplication-by-m map

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

is a degree-m^2 isogeny. If $m \neq 0$ in the base field, its kernel is

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

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Example #4: For $E/\mathbb{F}_q$, the map

$$\pi\colon (x, y) \mapsto (x^q, y^q)$$

is a degree-q isogeny, the *Frobenius endomorphism*.

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Example #4: For $E/\mathbb{F}_q$, the map

$$\pi : (x, y) \mapsto (x^q, y^q)$$

is a degree-q isogeny, the *Frobenius endomorphism*.

The kernel of $\pi - 1$ is precisely the set of rational points $E(\mathbb{F}_q)$.

Isogenies (examples)

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

- given by rational functions.
- a group homomorphism.

Example #4: For $E/\mathbb{F}_q$, the map

$$\pi\colon (x,y) \mapsto (x^q, y^q)$$

is a degree-q isogeny, the *Frobenius endomorphism*.

The kernel of $\pi - 1$ is precisely the set of rational points $E(\mathbb{F}_q)$.
Important fact: An isogeny φ is $\mathbb{F}_q$-rational iff $\pi \circ \varphi = \varphi \circ \pi$.

In SageMath:

```
sage: E = EllipticCurve(GF(101), [1,0])
sage: mu = E.scalar_multiplication(5)
```

In SageMath:

```
sage: E = EllipticCurve(GF(101), [1,0])
sage: mu = E.scalar_multiplication(5)
sage: mu
Scalar-multiplication endomorphism [5]
    of Elliptic Curve defined by y^2 = x^3 + x
    over Finite Field of size 101
```

In SageMath:

```
sage: E = EllipticCurve(GF(101), [1,0])
sage: mu = E.scalar_multiplication(5)
sage: mu
Scalar-multiplication endomorphism [5]
    of Elliptic Curve defined by y^2 = x^3 + x
    over Finite Field of size 101
sage: mu.rational_maps()
((x^25 + x^23 + ... + 14*x^3 + 25*x)
    /(25*x^24 + 14*x^22 - ... + x^2 + 1),
 (50*x^36*y + 20*x^34*y + ... + 45*x^2*y + 48*y)
    /(-12*x^36 - 2*x^34 + ... - 26*x^2 + 50))
```

The isogeny relation

Isogenies between distinct curves are "rare".

We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

The isogeny relation

Isogenies between distinct curves are "rare".

We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi\colon E \to E'$ has a unique dual isogeny $\widehat{\varphi}\colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

The isogeny relation

Isogenies between distinct curves are "rare".

We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi\colon E \to E'$ has a unique dual isogeny $\widehat{\varphi}\colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

<u>Tate's theorem</u>:
$E, E'/\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

(The Schoof–Elkies–Atkin algorithm can compute $\#E(\mathbb{F}_q)$ efficiently!)

The isogeny relation

Isogenies between distinct curves are "rare".

We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi \colon E \to E'$ has a unique dual isogeny $\widehat{\varphi} \colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

Tate's theorem:

$E, E'/\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

(The Schoof–Elkies–Atkin algorithm can compute $\#E(\mathbb{F}_q)$ efficiently!)

$\Longrightarrow$ Bottom line: Being isogenous is an equivalence relation.
Over finite fields, we can easily test it.

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

[1] (up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G : E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G : E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
  (...but they are only efficient for "small" degrees!)

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
  (...but they are only efficient for "small" degrees!)

$\rightsquigarrow$ Decompose large-degree isogenies into prime steps.
That is: Walk in an isogeny graph.

[1](up to isomorphism of E')

In SageMath:

```
sage: E = EllipticCurve(GF(419), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
                                over Finite Field of size 419
sage: K = E(80,30)
sage: K.order()
7
```

In SageMath:

```
sage: E = EllipticCurve(GF(419), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
                                 over Finite Field of size 419
sage: K = E(80,30)
sage: K.order()
7
sage: phi = E.isogeny(K)
sage: phi
Isogeny of degree 7
    from Elliptic Curve defined by y^2 = x^3 + x
                                 over Finite Field of size 419
    to Elliptic Curve defined by y^2 = x^3 + 285*x + 87
                                 over Finite Field of size 419
```

In SageMath:

```
sage: E = EllipticCurve(GF(419), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
                              over Finite Field of size 419
sage: K = E(80,30)
sage: K.order()
7
sage: phi = E.isogeny(K)
sage: phi
Isogeny of degree 7
     from Elliptic Curve defined by y^2 = x^3 + x
                              over Finite Field of size 419
     to Elliptic Curve defined by y^2 = x^3 + 285*x + 87
                              over Finite Field of size 419
sage: phi(K)
(0 : 1 : 0)      # φ(K) = ∞  ⟹  K lies in the kernel
```

In SageMath:

```
sage: E = EllipticCurve(GF(419), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
                          over Finite Field of size 419
sage: K = E(80,30)
sage: K.order()
7
sage: phi = E.isogeny(K)
sage: phi
Isogeny of degree 7
    from Elliptic Curve defined by y^2 = x^3 + x
                          over Finite Field of size 419
    to Elliptic Curve defined by y^2 = x^3 + 285*x + 87
                          over Finite Field of size 419
sage: phi(K)
(0 : 1 : 0)      # φ(K) = ∞  ⟹  K lies in the kernel
sage: phi.rational_maps()
((x^7 + 129*x^6 - ... + 25)/(x^6 + 129*x^5 - ... + 36),
 (x^9*y - 16*x^8*y - ... + 70*y)/(x^9 - 16*x^8 + ...))
```

Isogeny graphs

Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The *S*-isogeny graph over k consists of

Isogeny graphs

Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The S-isogeny graph over k consists of

- vertices given by elliptic curves over k;

Isogeny graphs

Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The S-isogeny graph over k consists of

- vertices given by elliptic curves over k;
- edges given by ℓ-isogenies, $\ell \in S$, over k;

Isogeny graphs

Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The S-isogeny graph over k consists of

- vertices given by elliptic curves over k;
- edges given by ℓ-isogenies, $\ell \in S$, over k;

up to k-isomorphism.

Isogeny graphs

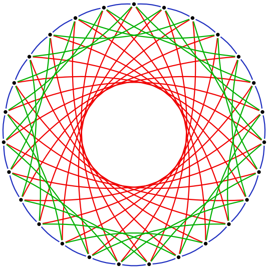
Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The S-isogeny graph over k consists of

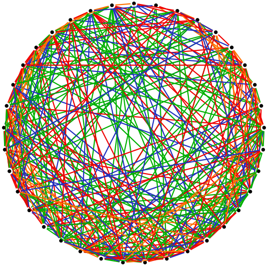
- vertices given by elliptic curves over k;
- edges given by ℓ-isogenies, $\ell \in S$, over k;

up to k-isomorphism.

Example components containing $E\colon y^2 = x^3 + x$:



$k = \mathbb{F}_{419}, \; S = \{3, 5, 7\}$



$k = \mathbb{F}_{431^2}, \; S = \{2, 3, 5, 7\}$.

Predictable groups

Elliptic curves in general can be very annoying

Predictable groups

Elliptic curves in general can be very annoying *computationally*:
Points in $E[\ell]$ have a tendency to live in large extension fields.

Predictable groups

Elliptic curves in general can be very annoying *computationally*: Points in $E[\ell]$ have a tendency to live in large extension fields.

Solution:

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *supersingular* if and only if $\#E(\mathbb{F}_p) = p+1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ and $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p+1) \times \mathbb{Z}/(p+1)$.

Predictable groups

Elliptic curves in general can be very annoying *computationally*: Points in $E[\ell]$ have a tendency to live in large extension fields.

Solution:

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *supersingular* if and only if $\#E(\mathbb{F}_p) = p+1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ and
  $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p+1) \times \mathbb{Z}/(p+1)$.

$\rightsquigarrow$ Easy method to control the group structure by choosing p!

$\rightsquigarrow$ Cryptography works well using supersingular curves.

Predictable groups

Elliptic curves in general can be very annoying *computationally*: Points in $E[\ell]$ have a tendency to live in large extension fields.

Solution:

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *supersingular* if and only if $\#E(\mathbb{F}_p) = p+1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ and
  $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p+1) \times \mathbb{Z}/(p+1)$.

$\rightsquigarrow$ Easy method to control the group structure by choosing p!

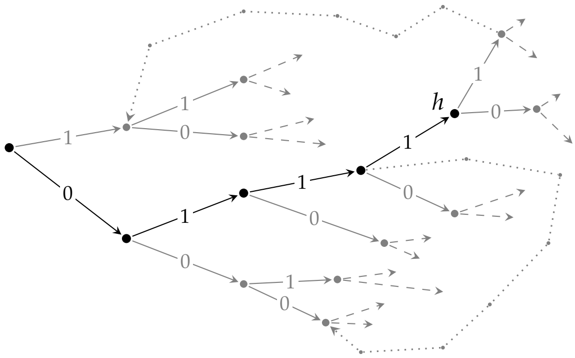
$\rightsquigarrow$ Cryptography works well using supersingular curves.

(All curves are supersingular until lunch time.)

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function.
- The CSIDH non-interactive key exchange.
- Hardness of isogeny problems, and reductions.
- The SQIsign signature scheme.
- Transcending to higher dimensions.

The Charles–Goren–Lauter hash function



- Start at some curve E.
- For each input digit b: Map the pair (E, b) to a finite subgroup $H \leq E$, compute $\varphi_H\colon E \to E'$, and set $E \leftarrow E'$.
- Finally return E.

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function. ✓
- The CSIDH non-interactive key exchange.
- Hardness of isogeny problems, and reductions.
- The SQIsign signature scheme.
- Transcending to higher dimensions.


CSIDH [ˈsiːˌsaɪd]
[Castryck–Lange–Martindale–Panny–Renes 2018]

Isogeny-based key exchange: High-level view

E

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E_A \\
E_B &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A \colon E \to E_A$ and $\varphi_B \colon E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E_A \arrow[dl, dashed, shift left] \\
E_B \arrow[ur, dashed, shift left] &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.

Isogeny-based key exchange: High-level view

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi_B'"] \arrow[ddll, dashed, bend left=5] \\
 & & E_{AB} \\
E_B \arrow[r, "\varphi_A'"'] \arrow[uurr, dashed, bend left=5] & E_{BA} &
\end{tikzcd}$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob <u>somehow</u> finds $\varphi_{B'} : E_A \to E_{AB}$,

Isogeny-based key exchange: High-level view

$$
\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \arrow[ddll, dashed, shift left] \\
& & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] \arrow[uurr, dashed, shift left] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}
$$

- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and
  Bob <u>somehow</u> finds $\varphi_{B'} : E_A \to E_{AB}$, such that $E_{AB} \cong E_{BA}$.

How to find "parallel" isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi'_B"] \\
& & E_{AB} \\
E_B \arrow[r, "\varphi'_A"'] & E_{BA} \arrow[ur, equal] &
\end{tikzcd}$$

How to find “parallel” isogenies?

$$\begin{tikzcd}
E \arrow[rr, "\varphi_A"] \arrow[dd, "\varphi_B"'] & & E_A \arrow[d, "\varphi_B'"] \\
& & E_{AB} \arrow[dl, equal] \\
E_B \arrow[r, "\varphi_A'"'] & E_{BA} &
\end{tikzcd}$$

CSIDH’s solution:

Use special isogenies φ_A which can be transported to the curve E_B totally independently of the secret isogeny φ_B.

(Similarly with reversed roles, of course.)

“Special” isogenies

Let $E/\mathbb{F}_p$ be supersingular and recall $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

"Special" isogenies

Let $E/\mathbb{F}_p$ be supersingular and recall $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

“Special” isogenies

Let $E/\mathbb{F}_p$ be supersingular and recall $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

$\rightsquigarrow$ For all such E can canonically find an isogeny $\varphi_\ell \colon E \to E'$.

“Special” isogenies

Let $E/\mathbb{F}_p$ be supersingular and recall $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ For every $\ell \mid (p+1)$ exists a unique order-ℓ subgroup H_ℓ.

$\rightsquigarrow$ For all such E can canonically find an isogeny $\varphi_\ell \colon E \to E'$.

We consider prime ℓ and refer to φ_ℓ as a “special” isogeny.

Cycles from "special" isogenies

What happens when we iterate such a "special" isogeny?

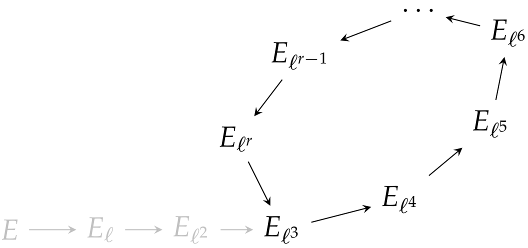
Cycles from “special” isogenies

What happens when we iterate such a “special” isogeny?

$$\begin{tikzcd}
 & & & E_{\ell^{r-1}} \arrow[d] & \cdots \arrow[l] & E_{\ell^6} \arrow[l] \\
 & & & E_{\ell^r} \arrow[d] & & E_{\ell^5} \arrow[u] \\
E \arrow[r] & E_{\ell} \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[r] & E_{\ell^4} \arrow[ur] &
\end{tikzcd}$$

Cycles from "special" isogenies

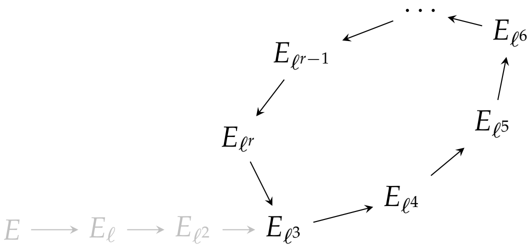
What happens when we iterate such a "special" isogeny?



- Exercise: Each curve has only one other rational ℓ-isogeny.

Cycles from "special" isogenies

What happens when we iterate such a "special" isogeny?



- Exercise: Each curve has only one other rational ℓ-isogeny.

‼ Reverse arrows are unique; the "tail" $E \to E_{\ell^3}$ cannot exist.

Cycles from "special" isogenies

What happens when we iterate such a "special" isogeny?

$$\begin{tikzcd}
 & & & E_{\ell^{r-1}} \arrow[d] & \cdots \arrow[l] & E_{\ell^6} \arrow[l] \\
 & & & E_{\ell^r} \arrow[d] & & \\
E \arrow[r] & E_{\ell} \arrow[r] & E_{\ell^2} \arrow[r] & E_{\ell^3} \arrow[r] & E_{\ell^4} \arrow[r] & E_{\ell^5} \arrow[uu]
\end{tikzcd}$$

- Exercise: Each curve has only one other rational ℓ-isogeny.

!! Reverse arrows are unique; the "tail" $E \to E_{\ell^3}$ cannot exist.

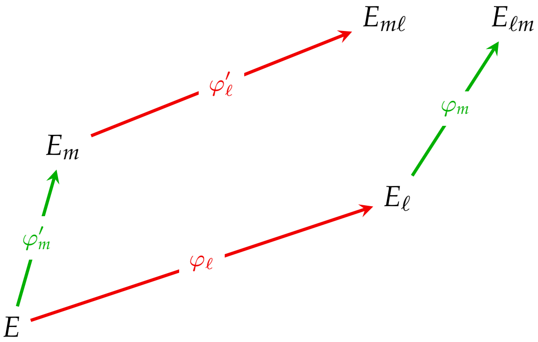
$\implies$ The "special" isogenies φ_ℓ form isogeny cycles!

Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?

Compatible cycles from “special” isogenies

What happens when we compose those “special” isogenies?



Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?

$$\begin{tikzcd}
 & E_{m\ell} & E_{\ell m} \\
E_m \arrow[ur, "\varphi'_\ell"] & E_\ell \arrow[ur, "\varphi_m"] & \\
E \arrow[u, "\varphi'_m"] \arrow[ur, "\varphi_\ell"'] & &
\end{tikzcd}$$

- Exercise: $\ker(\varphi'_\ell \circ \varphi'_m) = \ker(\varphi_m \circ \varphi_\ell) = \langle \ker \varphi_\ell, \ker \varphi'_m \rangle$.

Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?

$$\begin{tikzcd}
E_m \arrow[r, "\varphi'_\ell"] & E_{\ell m} \\
E \arrow[u, "\varphi'_m"] \arrow[r, "\varphi_\ell"] & E_\ell \arrow[u, "\varphi_m"']
\end{tikzcd}$$

- Exercise: $\ker(\varphi'_\ell \circ \varphi'_m) = \ker(\varphi_m \circ \varphi_\ell) = \langle \ker \varphi_\ell, \ker \varphi'_m \rangle$.

!! The order cannot matter $\implies$ cycles must be compatible.

CSIDH in one slide

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

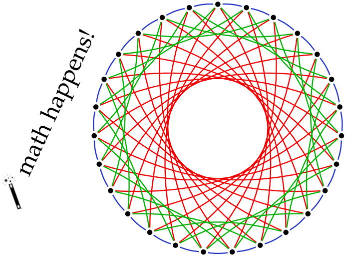
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.

CSIDH in one slide

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

CSIDH in one slide

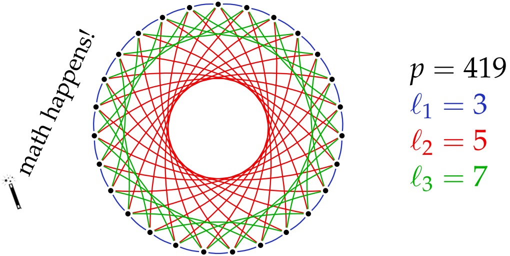
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.



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

CSIDH in one slide

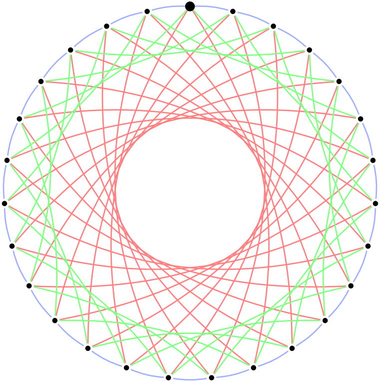
- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.



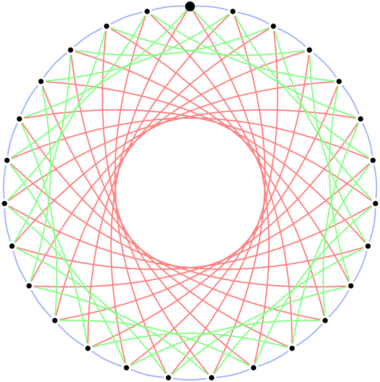
- Walking "left" and "right" on any ℓ_i-subgraph is efficient.

CSIDH key exchange

Alice
[**+**, **+**, **−**, **−**]



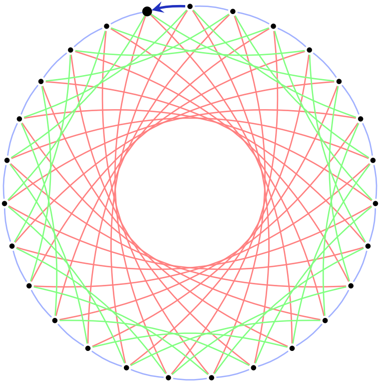
Bob
[**−**, **+**, **−**, **−**]



CSIDH key exchange

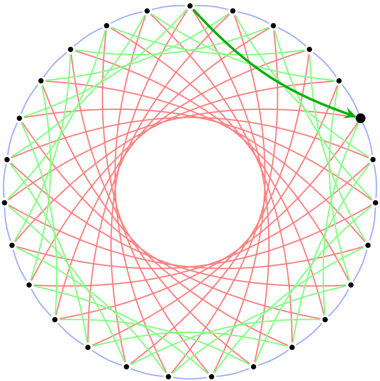
Alice

$[\mathbf{+}, \mathbf{+}, \mathbf{-}, \mathbf{-}]$
$\uparrow$



Bob

$[\mathbf{-}, \mathbf{+}, \mathbf{-}, \mathbf{-}]$
$\uparrow$



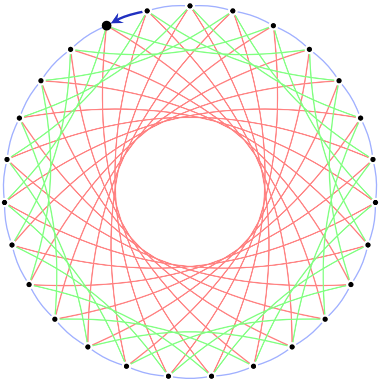
CSIDH key exchange

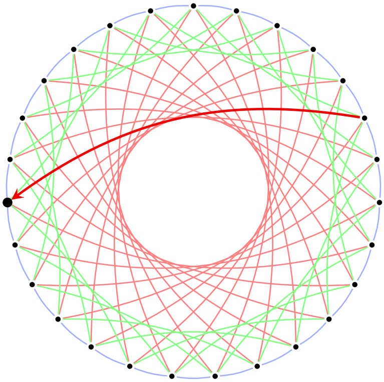
Alice

$[+, +, -, -]$
$\quad\ \uparrow$

Bob

$[-, +, -, -]$
$\quad\ \uparrow$

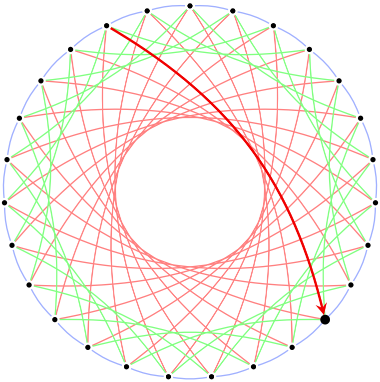




CSIDH key exchange

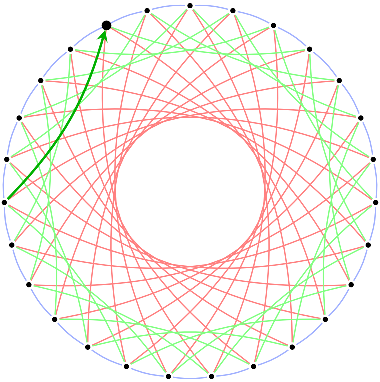
Alice

[+, +, −, −]



Bob

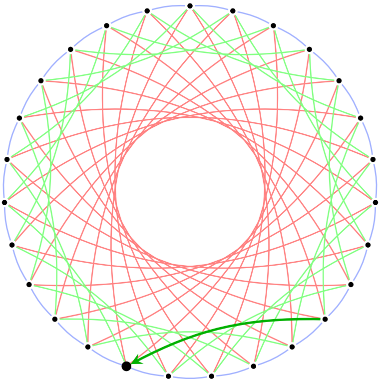
[−, +, −, −]



CSIDH key exchange

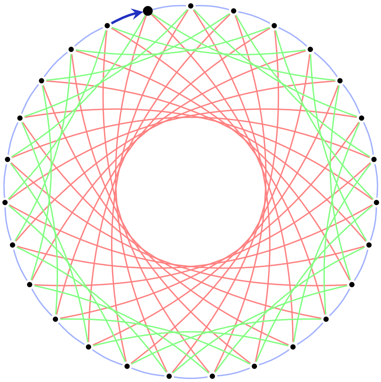
Alice

$[+, +, -, \underset{\uparrow}{-}]$



Bob

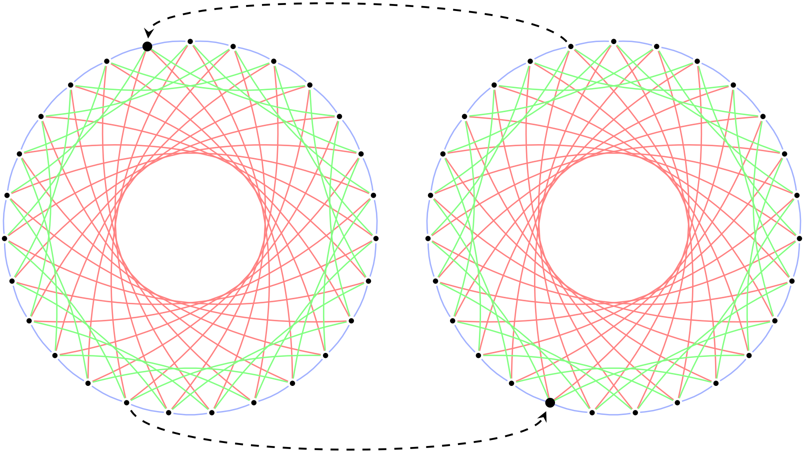
$[-, +, -, \underset{\uparrow}{-}]$



CSIDH key exchange

Alice
[+, +, −, −]

Bob
[−, +, −, −]



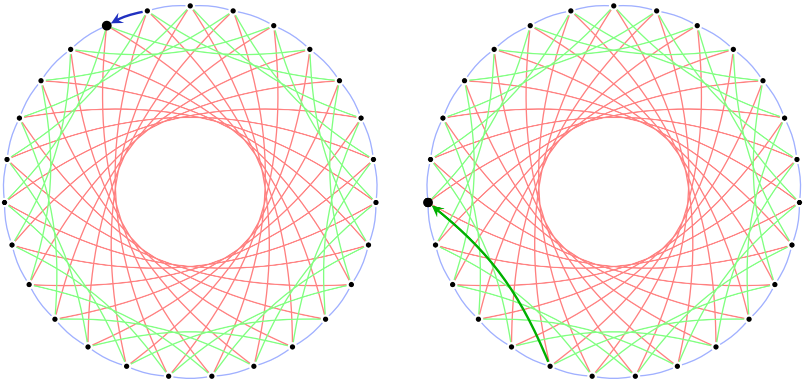
CSIDH key exchange

Alice

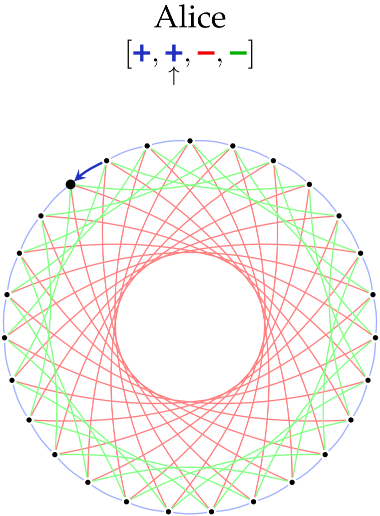
$[\underset{\uparrow}{+}, +, -, -]$

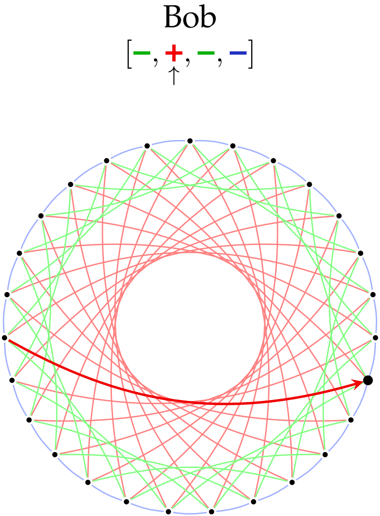
Bob

$[\underset{\uparrow}{-}, +, -, -]$



CSIDH key exchange

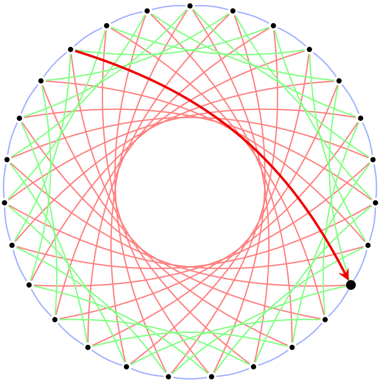




CSIDH key exchange

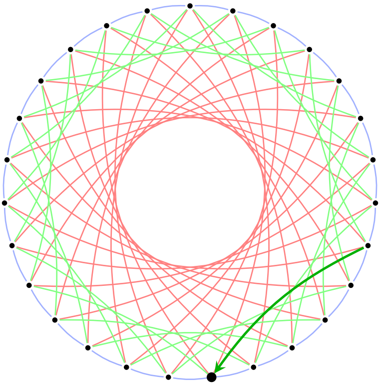
Alice

$[+, +, \underset{\uparrow}{-}, -]$



Bob

$[-, +, \underset{\uparrow}{-}, -]$



CSIDH key exchange

Alice

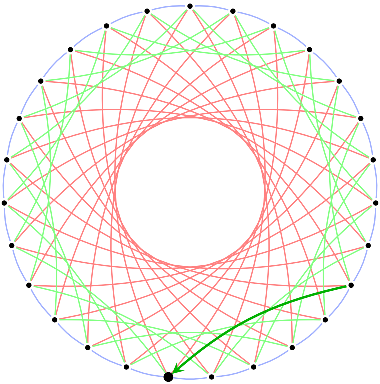
[+, +, −, −]

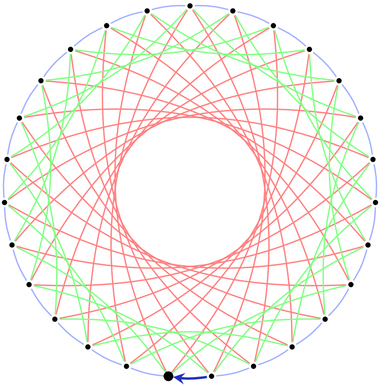
$\uparrow$

Bob

[−, +, −, −]

$\uparrow$

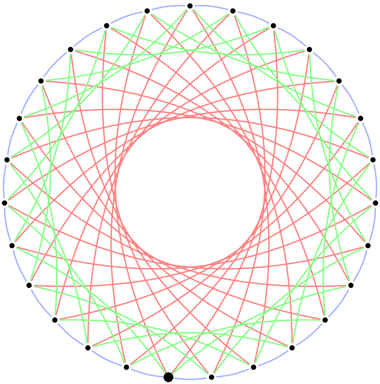




CSIDH key exchange

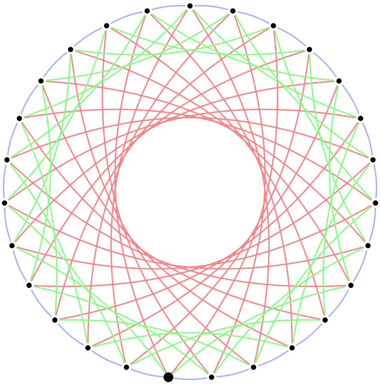
Alice

[+, +, −, −]



Bob

[−, +, −, −]



And... action! 🎬

Cycles are compatible: [right then left] = [left then right]

And... action!

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 $(+1, \quad 0, -3) \in \mathbb{Z}^3$.

And... action!

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 $(+1, \;\; 0, -3) \in \mathbb{Z}^3$.

There is a group action of $(\mathbb{Z}^n, +)$ on our set of curves X!

(An action of a group $(G, \cdot)$ on a set X is a map $*\colon G \times X \to X$
such that $id * x = x$ and $g * (h * x) = (g \cdot h) * x$ for all $g, h \in G$ and $x \in X$.)

The class group

<u>Recall</u>: Group action of $(\mathbb{Z}^n, +)$ on set of curves X.

The class group

Recall: Group action of $(\mathbb{Z}^n, +)$ on set of curves X.

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.

There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.

The class group

Recall: Group action of $(\mathbb{Z}^n, +)$ on set of curves X.

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.
There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.
Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

The class group

Recall: Group action of $(\mathbb{Z}^n, +)$ on set of curves X.

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.
There exist vectors $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.
Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

We understand the structure: By complex-multiplication theory, the quotient $\mathbb{Z}^n/\Lambda$ is the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

The class group

Recall: Group action of $(\mathbb{Z}^n, +)$ on set of curves X.

!! The set X is **finite** $\Longrightarrow$ The action is **not free**.
There exist vectors $\underline{v} \in \mathbb{Z}^n \setminus \{0\}$ which act trivially.
Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$.

We understand the structure: By complex-multiplication theory, the quotient $\mathbb{Z}^n/\Lambda$ is the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

!! This group characterizes *when two paths lead to the same curve*.

Walking in the CSIDH graph

- Recall: "Left" and "right" steps correspond to isogenies with special subgroups of E as kernels.

Walking in the CSIDH graph

- Recall: "Left" and "right" steps correspond to isogenies with special subgroups of E as kernels.

Computing a "left" step:

1. Find a point $(x, y) \in E$ of order ℓ_i with $x, y \in \mathbb{F}_p$.
2. Compute the isogeny with kernel $\langle (x, y) \rangle$.

Walking in the CSIDH graph

- Recall: "Left" and "right" steps correspond to isogenies with special subgroups of E as kernels.

Computing a "left" step:

1. Find a point $(x, y) \in E$ of order ℓ_i with $x, y \in \mathbb{F}_p$.
2. Compute the isogeny with kernel $\langle (x, y) \rangle$.

Computing a "right" step:

1. Find a point $(x, y) \in E$ of order ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
2. Compute the isogeny with kernel $\langle (x, y) \rangle$.

Walking in the CSIDH graph

- Recall: "Left" and "right" steps correspond to isogenies with special subgroups of E as kernels.

Computing a "left" step:

1. Find a point $(x, y) \in E$ of order ℓ_i with $x, y \in \mathbb{F}_p$.
2. Compute the isogeny with kernel $\langle (x, y) \rangle$.

Computing a "right" step:

1. Find a point $(x, y) \in E$ of order ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
2. Compute the isogeny with kernel $\langle (x, y) \rangle$.

(Finding a point of order ℓ_i: Pick $x \in \mathbb{F}_p$ random. Find $y \in \mathbb{F}_{p^2}$ such that $P = (x, y) \in E$. Compute $Q = [\frac{p+1}{\ell_i}]P$. Hope that $Q \neq \infty$, else retry.)

In SageMath:

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
          over Finite Field in z2 of size 419^2
```

In SageMath:

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
          over Finite Field in z2 of size 419^2
sage: while True:
....:     x = GF(419).random_element()
....:     try:
....:         P = E.lift_x(x)
....:     except ValueError: pass
....:     if P[1] in GF(419):   # "right" step: invert
....:         break
....:
sage: P
(218 : 403 : 1)
```

In SageMath:

```
sage: E = EllipticCurve(GF(419^2), [1,0])
sage: E
Elliptic Curve defined by y^2 = x^3 + x
        over Finite Field in z2 of size 419^2
sage: while True:
....:     x = GF(419).random_element()
....:     try:
....:         P = E.lift_x(x)
....:     except ValueError: pass
....:     if P[1] in GF(419):   # "right" step: invert
....:         break
....:
sage: P
(218 : 403 : 1)
sage: P.order().factor()
2 * 3 * 7
sage: EE = E.isogeny_codomain(2*3*P)  # "left" 7-step
sage: EE
Elliptic Curve defined by y^2 = x^3 + 285*x + 87
        over Finite Field in z2 of size 419^2
```

Efficient x-only arithmetic

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)

Efficient x-only arithmetic

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

Efficient x-only arithmetic

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\implies$ We get an induced map $\mathtt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \mathtt{xMUL}_n(x(P)) = x([n]P)\,.$$

Efficient x-only arithmetic

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\Longrightarrow$ We get an induced map $\texttt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \texttt{xMUL}_n(x(P)) = x([n]P)\,.$$

The same reasoning works for isogeny formulas.

Net result: With x-only arithmetic everything happens over $\mathbb{F}_p$.
$\Longrightarrow$ (Relatively) efficient CSIDH implementations!

Why no Shor?

Shor's quantum algorithm computes α from g^α in any group in polynomial time.

Why no Shor?

Shor's quantum algorithm computes α from g^α in any group in polynomial time.

Shor computes α from $h = g^\alpha$ by finding the kernel of the map

$$f\colon\ \mathbb{Z}^2 \to G,\ (x, y) \mapsto g^x \cdot h^y\,.$$

Why no Shor?

Shor's quantum algorithm computes α from g^α in any group in polynomial time.

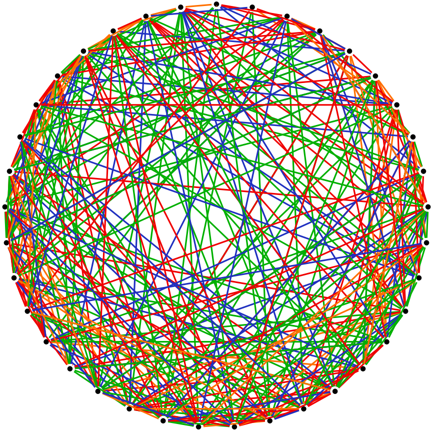
Shor computes α from $h = g^\alpha$ by finding the kernel of the map

$$f\colon\ \mathbb{Z}^2 \to G,\ (x, y) \mapsto g^x \underset{\uparrow}{\cdot} h^y\,.$$

For group <u>actions</u>, we simply cannot compose $a * s$ and $b * s$!

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function. ✓
- The CSIDH non-interactive key exchange. ✓
- Hardness of isogeny problems, and reductions.
- The SQIsign signature scheme.
- Transcending to higher dimensions.



Now:

Supersingular isogeny graphs over $\mathbb{F}_{p^2}$.

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find any isogeny $E \to E'$.

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find <u>any</u> isogeny $E \to E'$.

<u>Fact</u>: The supersingular isogeny graph has size $\lfloor p/12 \rfloor + \varepsilon$.

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find any isogeny $E \to E'$.

Fact: The supersingular isogeny graph has size $\lfloor p/12 \rfloor + \varepsilon$.

Classical attacks:

- Meet-in-the-middle: $\tilde{O}(p^{1/2})$ time & *space* ***(!)***.

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find any isogeny $E \to E'$.

Fact: The supersingular isogeny graph has size $\lfloor p/12 \rfloor + \varepsilon$.

Classical attacks:

- Meet-in-the-middle: $\tilde{O}(p^{1/2})$ time *& space (!)*.
- Delfs–Galbraith: $\tilde{O}(p^{1/2})$ time, negligible space.

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find any isogeny $E \to E'$.

Fact: The supersingular isogeny graph has size $\lfloor p/12 \rfloor + \varepsilon$.

Classical attacks:

- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/2})$ time & *space* (*!*).
- Delfs–Galbraith: $\tilde{\mathcal{O}}(p^{1/2})$ time, negligible space.

Quantum attacks:

- Biasse–Jao–Sankar: $\tilde{\mathcal{O}}(p^{1/4})$. (Quantum version of Delfs–Galbraith.)

The supersingular isogeny problem

Most contemporary isogeny-based cryptography reduces to:

***The* supersingular isogeny problem.** Given two supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$, find <u>any</u> isogeny $E \to E'$.

<u>Fact</u>: The supersingular isogeny graph has size $\lfloor p/12 \rfloor + \varepsilon$.

<u>Classical</u> attacks:

- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/2})$ time & *space (!)*.
- Delfs–Galbraith: $\tilde{\mathcal{O}}(p^{1/2})$ time, negligible space.

<u>Quantum</u> attacks:

- Biasse–Jao–Sankar: $\tilde{\mathcal{O}}(p^{1/4})$. (Quantum version of Delfs–Galbraith.)

<u>Bottom line</u>: Fully exponential. Complexity $\exp\big((\log p)^{1+o(1)}\big)$.

The endomorphism-ring problem

Most contemporary isogeny-based cryptography reduces to:

The supersingular endomorphism-ring problem.
For a supersingular elliptic curve, find its endomorphism ring.

The endomorphism-ring problem

Most contemporary isogeny-based cryptography reduces to:

The supersingular endomorphism-ring problem.
For a supersingular elliptic curve, find its endomorphism ring.

Attacks:

- Nontrivial endomorphisms are "just" self-isogenies.

$\rightsquigarrow$ Dominating cost: Find cycles in isogeny graphs.

$\rightsquigarrow$ Algorithms are morally similar to the isogeny problem, followed by a polynomial-time post-processing phase.

The endomorphism-ring problem

Most contemporary isogeny-based cryptography reduces to:

The supersingular endomorphism-ring problem.
For a supersingular elliptic curve, find its endomorphism ring.

Attacks:

- Nontrivial endomorphisms are "just" self-isogenies.

$\rightsquigarrow$ Dominating cost: Find cycles in isogeny graphs.

$\rightsquigarrow$ Algorithms are morally similar to the isogeny problem, followed by a polynomial-time post-processing phase.

Theorem (Wesolowski 2021): Assuming GRH, the isogeny and endomorphism-ring problems are *polynomial-time equivalent*.

SoK: Isogeny problems

Some isogeny problems are much more broken than others.

SoK: Isogeny problems

Some isogeny problems are much more broken than others.

Is SIKE broken yet?

Home About

Schemes

Name	Type	Classical Security	Quantum Security	References	Additional Information
▹ SIDH	Key Exchange	$\tilde{O}(n^3)$	$\tilde{O}(n^3)$	JDF11 DJP14 CLN16	▹ Comment
SIKE	KEM	$\tilde{O}(n^3)$	$\tilde{O}(n^3)$	SIKE	▹ Comment
B-SIDH	Key Exchange	$\tilde{O}(n^3)$	$\tilde{O}(n^3)$	Cos19	▹ Comment
CRS	Key Exchange, Non Interactive Key Exchange	$\exp(n)^{1/2}$	L(1/2)	Cou06 RS06 DKS18	▹ Comment
CSIDH	Key Exchange, Non Interactive Key Exchange	$\exp(n)^{1/2}$	L(1/2)	CL+18 CD19	▹ Comment

https://issikebrokenyet.github.io

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function. ✓
- The CSIDH non-interactive key exchange. ✓
- Hardness of isogeny problems, and reductions. ✓
- The SQIsign signature scheme.
- Transcending to higher dimensions.

More “special” isogenies

- Earlier: “Special” isogenies φ_ℓ with rational kernel points.

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Recall the Frobenius endomorphism $\pi \colon (x, y) \mapsto (x^p, y^p)$.)

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Recall the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$.

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Recall the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

$\boxed{\text{Every isogeny } \varphi\colon E \to E' \text{ comes from a subset } I_\varphi \subseteq \mathrm{End}(E).}$

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Recall the Frobenius endomorphism $\pi \colon (x, y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota \colon (x, y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

Every isogeny $\varphi \colon E \to E'$ comes from a subset $I_\varphi \subseteq \mathrm{End}(E)$.

:) We understand the structure of $\mathrm{End}(E)$.

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Recall the Frobenius endomorphism $\pi\colon (x,y) \mapsto (x^p, y^p)$.)

!! Over $\mathbb{F}_{p^2}$, we can have more endomorphisms.
Example: $y^2 = x^3 + x$ has $\iota\colon (x,y) \mapsto (-x, \sqrt{-1} \cdot y)$.

- Extremely non-obvious fact in this setting:

Every isogeny $\varphi\colon E \to E'$ comes from a subset $I_\varphi \subseteq \mathrm{End}(E)$.

:) We understand the structure of $\mathrm{End}(E)$.

:) We understand how I_φ, I_ψ relate for isogenies $\varphi, \psi\colon E \to E'$.
(NB: Same E'.)

The Deuring correspondence

...is the formal version of what I just said.

The Deuring correspondence

...is the formal version of what I just said.

...is a strong connection between two *a priori* very different worlds:

The Deuring correspondence

...is the formal version of what I just said.

...is a strong connection between two *a priori* very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.

The Deuring correspondence

...is the formal version of what I just said.

...is a strong connection between two *a priori* very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

The Deuring correspondence

...is the formal version of what I just said.

...is a strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become "connecting ideals" in quaternion land.

The Deuring correspondence

...is the formal version of what I just said.

...is a strong connection between two *a priori* very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become "connecting ideals" in quaternion land.

$\ddot\smile$ One direction is easy, the other seems hard! $\rightsquigarrow$ *Cryptography!*

The Deuring correspondence (examples)

Let $p = 7799999$ and let $\mathbf{i}, \mathbf{j}$ satisfy $\mathbf{i}^2 = -1,\ \mathbf{j}^2 = -p,\ \mathbf{ji} = -\mathbf{ij}$.

The ring $\mathcal{O}_0 = \mathbb{Z} \oplus \mathbb{Z}\,\mathbf{i} \oplus \mathbb{Z}\,\frac{\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{1+\mathbf{ij}}{2}$
corresponds to the curve $E_0\colon\ y^2 = x^3 + x$.

The ring $\mathcal{O}_1 = \mathbb{Z} \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+32631010\mathbf{i}+\mathbf{ij}}{9894}$
corresponds to the curve $E_1\colon\ y^2 = x^3 + 1$.

The ideal $I = \mathbb{Z}\,4947 \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{598+4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+598\mathbf{i}+\mathbf{ij}}{2}$
defines an isogeny $E_0 \to E_1$ of degree $4947 = 3 \cdot 17 \cdot 97$.

Signing with isogenies

- Fiat–Shamir: signature scheme from identification scheme.

$$E_0 \xrightarrow{\text{secret}} E_A$$

Signing with isogenies

- <u>Fiat–Shamir</u>: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[rr, dashed, "secret"] \arrow[d, "commitment"'] & & E_A \\
E_1 & &
\end{tikzcd}$$

Signing with isogenies

- <u>Fiat–Shamir</u>: signature scheme from identification scheme.

$$
\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, "\textit{commitment}"'] & E_A \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}
$$

Signing with isogenies

- <u>Fiat–Shamir</u>: signature scheme from identification scheme.

$$
\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, "\textit{commitment}"'] & E_A \arrow[d, "\textit{signature}"] \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}
$$

Signing with isogenies

- Fiat–Shamir: signature scheme from identification scheme.

$$
\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, "commitment"'] & E_A \arrow[d, thick, "signature"] \\
E_1 \arrow[r, "challenge"'] & E_2
\end{tikzcd}
$$

- Easy signature: $E_A \to E_0 \to E_1 \to E_2$. *Obviously broken.*

Signing with isogenies

- <u>Fiat–Shamir</u>: signature scheme from identification scheme.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, "\textit{commitment}"'] & E_A \arrow[d, "\textit{signature}"] \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}$$

- Easy signature: $E_A \to E_0 \to E_1 \to E_2$. *Obviously broken.*
- <u>**SQIsign**'s solution</u>: Construct new path $E_A \to E_2$ (using *secret*).

SQIsign

Main idea:

- Construct the "signature square" in quaternion land.
- Project the secret and signature down to the curve world.
- The verifier can check on curves that everything is correct.

SQIsign

Main idea:

- Construct the "signature square" in quaternion land.
- Project the secret and signature down to the curve world.
- The verifier can check on curves that everything is correct.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.
- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

SQIsign

<u>Main idea:</u>

- Construct the "signature square" in quaternion land.
- Project the secret and signature down to the curve world.
- The verifier can check on curves that everything is correct.

<u>Main technical tool:</u> The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.
- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

$\rightsquigarrow$ SQIsign takes the "broken" signature $E_A \to E_0 \to E_1 \to E_2$ and rewrites it into a random isogeny $E_A \to E_2$.

SQIsign

Main idea:

- Construct the "signature square" in quaternion land.
- Project the secret and signature down to the curve world.
- The verifier can check on curves that everything is correct.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can find *smooth* isogeny $E \to E'$.
- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

$\rightsquigarrow$ SQIsign takes the "broken" signature $E_A \to E_0 \to E_1 \to E_2$ and rewrites it into a random isogeny $E_A \to E_2$.

> *"If you have KLPT implemented very nicely as a black box, then anyone can implement SQIsign."*
> — Yan Bo Ti

SQIsign: Numbers

sizes

parameter set	public keys	signatures
NIST-**I**	**64** bytes	**177** bytes
NIST-**III**	**96** bytes	**263** bytes
NIST-**V**	**128** bytes	**335** bytes

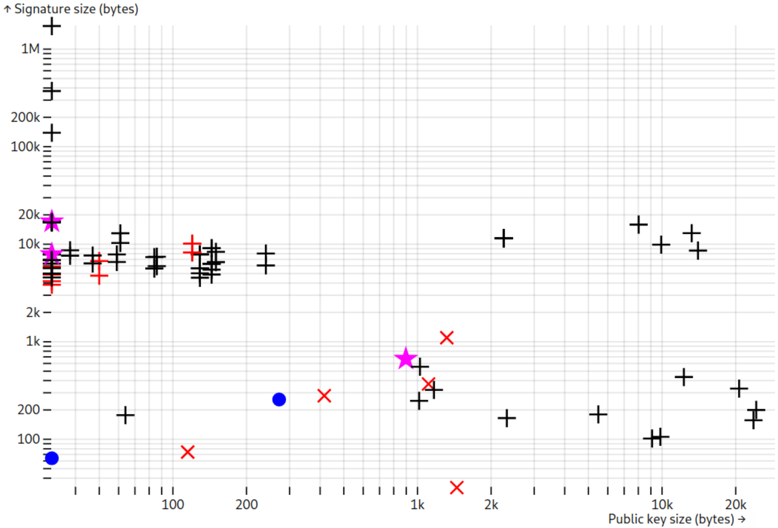
performance

Cycle counts for a *generic C implementation* running on an Intel *Ice Lake* CPU.
Optimizations are certainly possible and work in progress.

parameter set	keygen	signing	verifying
NIST-**I**	**3728** megacycles	**5779** megacycles	**108** megacycles
NIST-**III**	**23734** megacycles	**43760** megacycles	**654** megacycles
NIST-**V**	**91049** megacycles	**158544** megacycles	**2177** megacycles

Source: `https://sqisign.org`

SQIsign: Comparison



Source: `https://pqshield.github.io/nist-sigs-zoo`

SQIsign verification

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K \rangle$.

SQIsign verification

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K \rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

SQIsign verification

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K \rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

SQIsign verification

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K \rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(n^2 \cdot \ell)$.
Exponentially smaller than a ℓ^n-isogeny!

SQIsign verification

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K\rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(n^2 \cdot \ell)$.
Exponentially smaller than a ℓ^n-isogeny!

- Graph view: Each ψ_i is a step in the ℓ-isogeny graph.

In SageMath:

```
sage: E = EllipticCurve(GF(2^127-1), [1,0])
sage: K = E(23, 40490046516039691075571867486180936666)
sage: K.order()
10633823966279326983230456482242756608
sage: K.order().factor()
2^123
```

In SageMath:

```
sage: E = EllipticCurve(GF(2^127-1), [1,0])
sage: K = E(23, 4049004651603969107557186748618093666)
sage: K.order()
10633823966279326983230456482242756608
sage: K.order().factor()
2^123
sage: phi = E.isogeny(K, algorithm="factored")
sage: phi
Composite morphism of degree 1063...6608 = 2^123:
  From: Elliptic Curve defined by y^2 = x^3 + x
                over Finite Field of size 1701...5727
  To:   Elliptic Curve defined by
                y^2 = x^3 + 1625...8575*x + 1200...7360
                over Finite Field of size 1701...5727
```

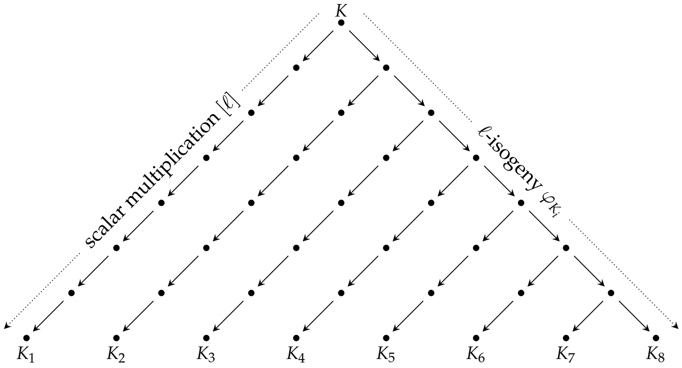
Strategies for composite-degree isogenies

Recall: We split ℓ^n-isogenies into n individual ℓ-isogenies ψ_i.
This requires computing $K_i := [\ell^{n-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(n)$ for all i.

Strategies for composite-degree isogenies

Recall: We split ℓ^n-isogenies into n individual ℓ-isogenies ψ_i.
This requires computing $K_i := [\ell^{n-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(n)$ for all i.

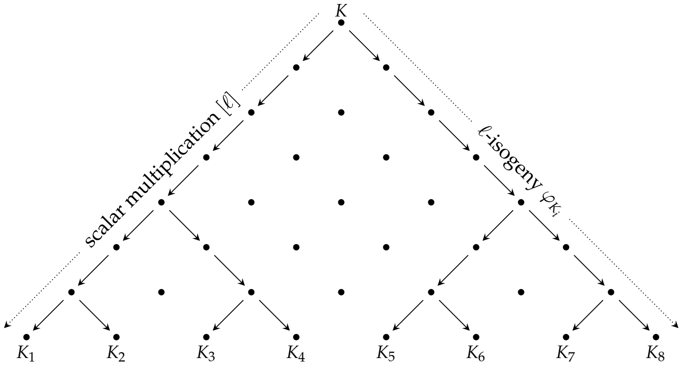
Naïve strategy:



Strategies for composite-degree isogenies

Recall: We split ℓ^n-isogenies into n individual ℓ-isogenies ψ_i.
This requires computing $K_i := [\ell^{n-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(n)$ for all i.

Sparse strategy:



Optimal strategies for composite-degree isogenies

$\Longrightarrow$ Sparse strategy improves $O(n^2 \cdot \ell)$ to $O(n \log n \cdot \ell)$.

Optimal strategies for composite-degree isogenies

$\Longrightarrow$ Sparse strategy improves $O(n^2 \cdot \ell)$ to $O(n \log n \cdot \ell)$.

When the costs of $[\ell]$ and φ_{K_i} are imbalanced, other trees can be even more efficient. They can be constructed easily.

$\rightsquigarrow$ *"optimal strategies"*

Optimal strategies for composite-degree isogenies

$\Longrightarrow$ Sparse strategy improves $O(n^2 \cdot \ell)$ to $O(n \log n \cdot \ell)$.

When the costs of $[\ell]$ and φ_{K_i} are imbalanced, other trees can be even more efficient. They can be constructed easily.

$\rightsquigarrow$ *"optimal strategies"*

Similar techniques exist for general *composite* degree.

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function. ✓
- The CSIDH non-interactive key exchange. ✓
- Hardness of isogeny problems, and reductions. ✓
- The SQIsign signature scheme. ✓
- Transcending to higher dimensions.

Gluing elliptic curves

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Gluing elliptic curves

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies of *products* of elliptic curves

Gluing elliptic curves

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies of *products* of elliptic curves

- The product $E \times E'$ is an abelian *surface*.
  Compare: A product of two lines is a plane!

Gluing elliptic curves

- Fallout from the SIDH attack: New tools.

"One man's a-ttack is another man's a-treasure."

Main technique underlying attack:

Computing isogenies of *products* of elliptic curves

- The product $E \times E'$ is an abelian *surface*.
  Compare: A product of two lines is a plane!
- Similar to elliptic curves in many ways:
  - Points form an abelian group.
  - Similar group structure, but more components.
  - Can define isogenies from kernel subgroups.

The embedding lemma

- Fallout from the SIDH attack: New tools.

The embedding lemma

- Fallout from the SIDH attack: New tools.

2.1. **The embedding lemma.** If α_1, α_2 are two endomorphisms of an elliptic curve E of degree a_1 and a_2, then $\alpha_1 \circ \alpha_2$ is of degree $a_1 a_2$. However it is harder to control the degree of the sum; by Cauchy-Schwartz we can bound it as: $(a_1^{1/2} - a_2^{1/2})^2 \leq \deg(\alpha_1 + \alpha_2) \leq (a_1^{1/2} + a_2^{1/2})^2$ (unless $\alpha_1 = -\alpha_2$). And $\alpha_1 + \alpha_2$ is of degree $a_1 + a_2$ if and only if $\alpha_1 \tilde{\alpha}_2$ is of trace 0.

If α_1 commutes with α_2, we can instead use Kani's lemma [Kan97, § 2] to build an endomorphism F *in dimension* 2 on E^2 which is an $(a_1 + a_2)$-isogeny (so is of degree $(a_1 + a_2)^2$ since we are in dimension 2). So by going to higher dimension we can combine degrees additively. The proof of this lemma is very simple (a simple two by two matrix computation), but its powerful algorithmic potential went unnoticed until Castrick and Decru applied it in [CD22] to attack on SIDH.

—Damien Robert [ePrint 2022/1704]

The embedding lemma

Consider a commutative diagram of isogenies

$$\begin{tikzcd} E \arrow[r, "\varphi"] \arrow[d, "\psi"'] & E' \arrow[d, "\psi'"] \\ E'' \arrow[r, "\varphi'"'] & E''' \end{tikzcd}$$

where $a := \deg \varphi$ and $b := \deg \psi$ are coprime; let $N := a + b$.

Lemma. Then

$$F := \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix}$$

defines an N-isogeny $E \times E''' \to E' \to E''$.

Its kernel is $\ker(F) = \left\{ (\widehat{\varphi}(P), \psi'(P)) \mid P \in E'[N] \right\}$.

Representing $\varphi|_{E[N]}$

Recall: For embedding lemma, need to evaluate φ on $E[N]$.

$\rightsquigarrow$ Exponentially many points. ☹

Representing $\varphi|_{E[N]}$

Recall: For embedding lemma, need to evaluate φ on $E[N]$.

$\rightsquigarrow$ Exponentially many points. ☹

Clever trick:

- Fix basis (P, Q) of $E[N]$; compute $P' = \varphi(P)$ and $Q' = \varphi(Q)$.
- Notice that φ is a group homomorphism.

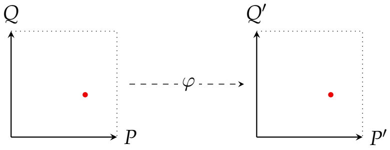
Representing $\varphi|_{E[N]}$

Recall: For embedding lemma, need to evaluate φ on $E[N]$.

$\rightsquigarrow$ Exponentially many points. ☹

Clever trick:

- Fix basis (P, Q) of $E[N]$; compute $P' = \varphi(P)$ and $Q' = \varphi(Q)$.
- Notice that φ is a group homomorphism.



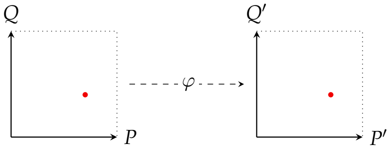
Representing $\varphi|_{E[N]}$

Recall: For embedding lemma, need to evaluate φ on $E[N]$.

$\rightsquigarrow$ Exponentially many points. ☹

Clever trick:

- Fix basis (P, Q) of $E[N]$; compute $P' = \varphi(P)$ and $Q' = \varphi(Q)$.
- Notice that φ is a group homomorphism.



Evaluating φ at an arbitrary point $T \in E[N]$:

1. Decompose $T = [u]P + [v]Q$ with $u, v \in \mathbb{Z}$.
   This is a DLP-like computation, which is easy whenever N is smooth!
2. Output $[u]P' + [v]Q'$.

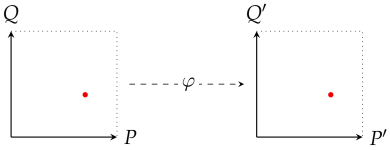
Representing $\varphi|_{E[N]}$

<u>Recall</u>: For embedding lemma, need to evaluate φ on $E[N]$.

$\rightsquigarrow$ Exponentially many points. ☹

<u>Clever trick</u>:

- Fix basis (P, Q) of $E[N]$; compute $P' = \varphi(P)$ and $Q' = \varphi(Q)$.
- Notice that φ is a group homomorphism.



<u>Evaluating φ at an arbitrary point $T \in E[N]$:</u>

1. Decompose $T = [u]P + [v]Q$ with $u, v \in \mathbb{Z}$.
   This is a DLP-like computation, which is easy whenever N is smooth!
2. Output $[u]P' + [v]Q'$.

$\Longrightarrow$ The data (P, Q, P', Q') encodes the *restriction* $\varphi|_{E[N]}$.

Plan for this lecture

- High-level overview for intuition. ✓
- Elliptic curves & isogenies. ✓
- The CGL hash function. ✓
- The CSIDH non-interactive key exchange. ✓
- Hardness of isogeny problems, and reductions. ✓
- The SQIsign signature scheme. ✓
- Transcending to higher dimensions. ✓

Ad

https://isogeny.club

Questions?

(Also feel free to email me: `lorenz@yx7.cc`)