Isogeny Group Actions

Lorenz Panny

Technische Universität München

ECC 2024 Autumn School, Taipei, 29 October 2024

Crypto(graphy) 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:

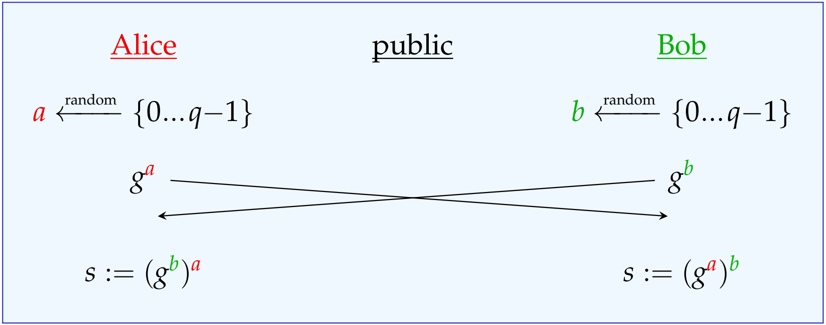
- a finite group G (traditionally $\mathbb{F}_p^*$, today elliptic curves)
- an element $g \in G$ of prime order q

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \ldots q-1\}$		$b \xleftarrow{\text{random}} \{0 \ldots q-1\}$
g^a	$g^a \rightarrow$ Bob, $g^b \rightarrow$ Alice	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



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

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 $B \leftarrow t \cdot g$.

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

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 $b + 2$.

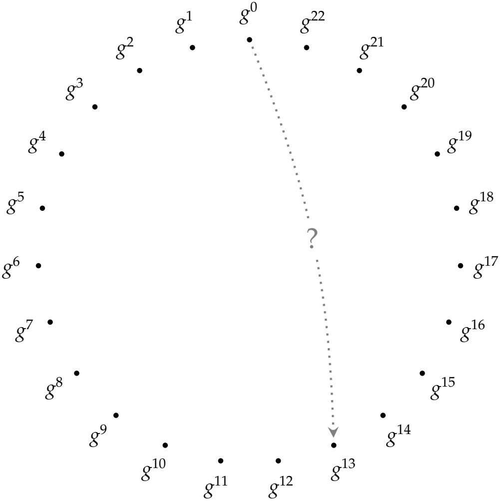
...

Diffie–Hellman: Bob vs. Eve

Bob	Attacker Eve
1. Set $t \leftarrow g$.	1. Set $t \leftarrow g$. If $t = B$ return 1.
2. Set $t \leftarrow t \cdot g$.	2. Set $t \leftarrow t \cdot g$. If $t = B$ return 2.
3. Set $t \leftarrow t \cdot g$.	3. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.
4. Set $t \leftarrow t \cdot g$.	4. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.
...	...
$b-2$. Set $t \leftarrow t \cdot g$.	$b-2$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b-2$.
$b-1$. Set $t \leftarrow t \cdot g$.	$b-1$. Set $t \leftarrow t \cdot g$. If $t = B$ return $b-1$.
b. Publish $B \leftarrow t \cdot g$.	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 $b+2$.
	...

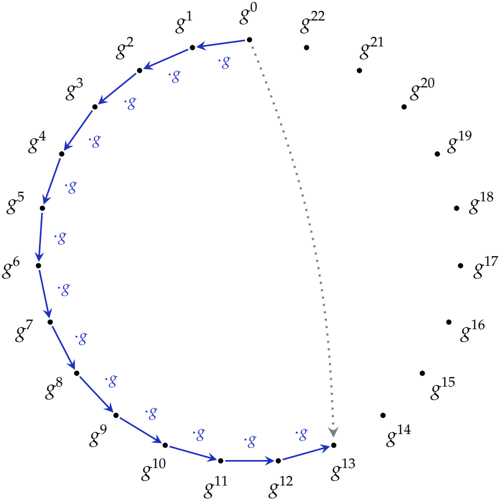
Effort for both: $O(\#G)$. Bob needs to be smarter.

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



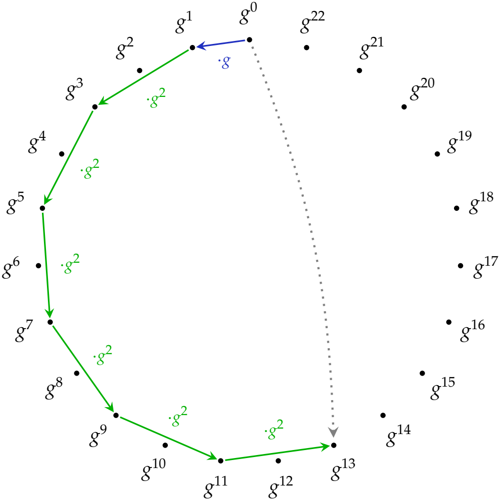
Bob computes his public key g^{13} from g.

multiply



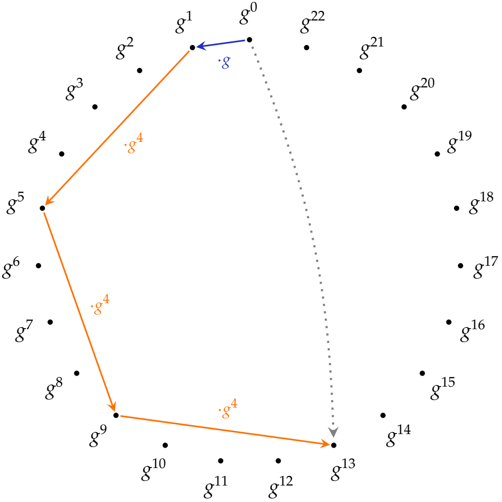
Bob computes his public key g^{13} from g.

Square-and-multiply



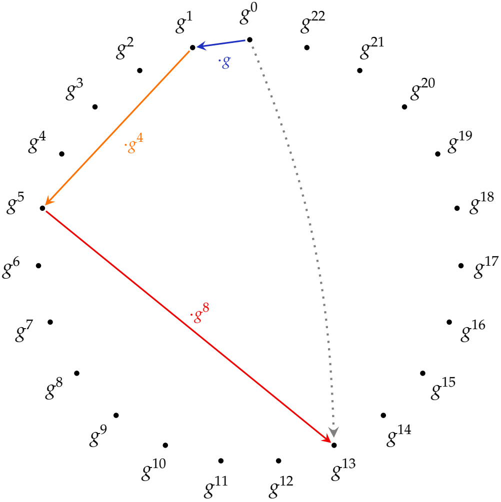
Bob computes his public key g^{13} from g.

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



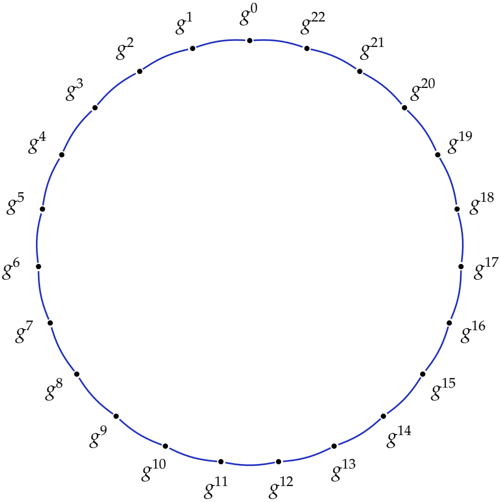
Bob computes his public key g^{13} from g.

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

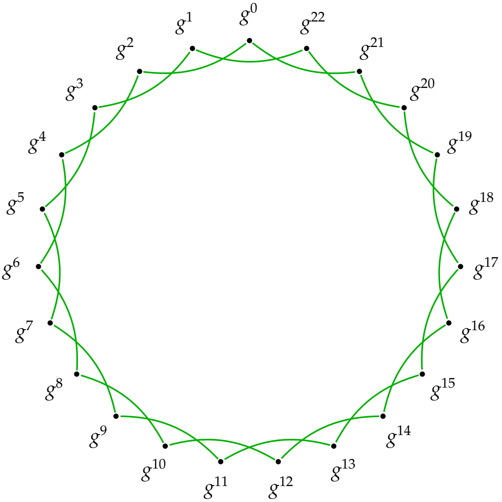


Bob computes his public key g^{13} from g.

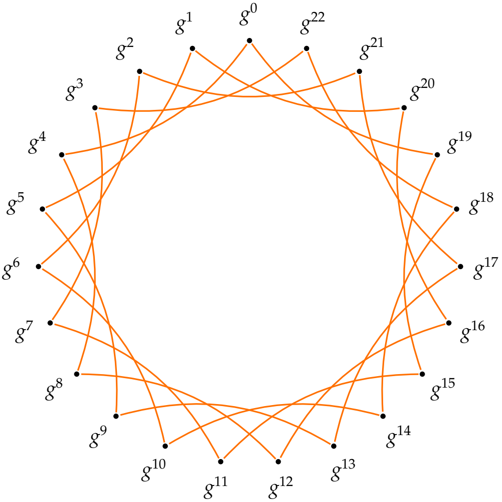
Square-and-multiply as graphs



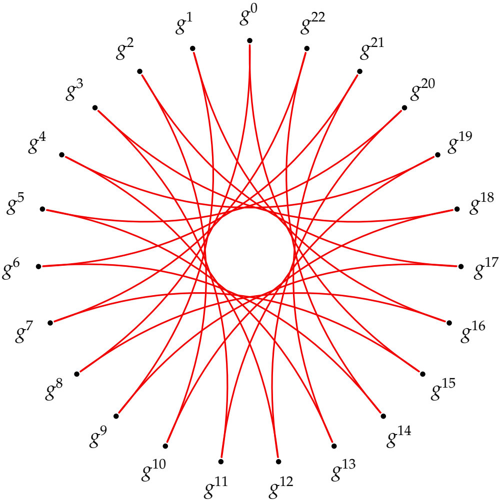
Square-and-multiply as graphs



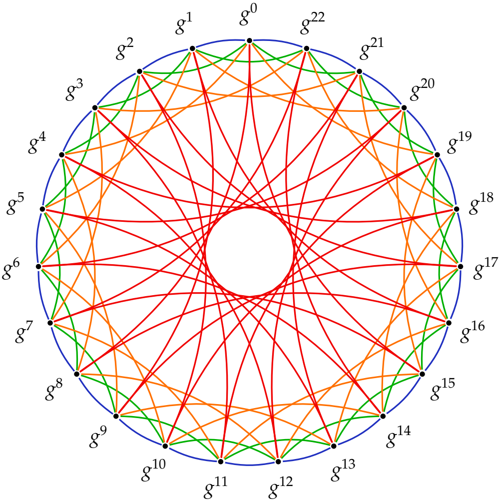
Square-and-multiply as graphs



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(# 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,

isogeny graphs

can replace DLP-based constructions post-quantumly.

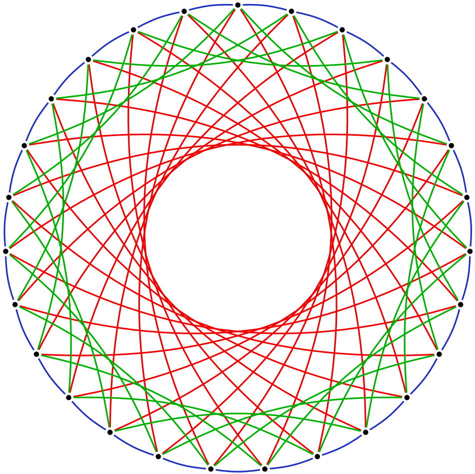
In some cases,

isogeny graphs

can replace DLP-based constructions post-quantumly.

some

Components of particular isogeny graphs look like this:



Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies.
- The CSIDH non-interactive key exchange.
- Classical and quantum security of CSIDH.
- Orientations and the SCALLOP family.
- *Un*restricted effective group actions.

Recap: Isogenies of elliptic curves

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

Recap: Isogenies of elliptic curves

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

- given by rational functions.

Recap: Isogenies of elliptic curves

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

- given by rational functions.
- a group homomorphism.

Recap: Isogenies of elliptic curves

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 $\varphi(P + Q) = \varphi(P) + \varphi(Q)$.

Recap: Isogenies of elliptic curves

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 each $m \neq 0$, the multiplication-by-m map

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

is a degree-m^2 isogeny.

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 $E/\mathbb{F}_q$, 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 #3: 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 #3: 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))
```

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.

[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 $\varphi_G \colon E \to E'$ 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.

[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.

[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.

⇝ 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!)

⇝ 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

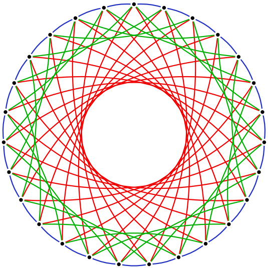
Consider a field k and let $S \not\ni \mathrm{char}(k)$ be a set of primes.

The S-isogeny graph over k consists of

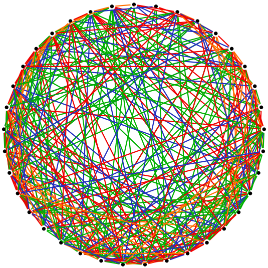
- vertices given by elliptic curves over k;
- edges given by ℓ-isogenies, $\ell \in S$, over k;

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.

<u>Solution:</u>

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is <u>*supersingular*</u> 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 about 14:00.)

Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies. ✓
- The CSIDH non-interactive key exchange.
- Classical and quantum security of CSIDH.
- Orientations and the SCALLOP family.
- *Un*restricted effective group actions.

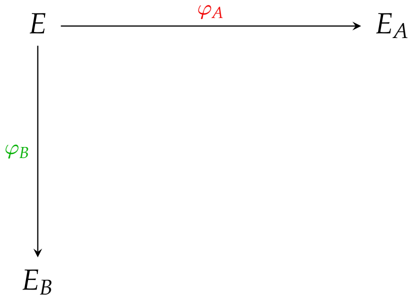
CSIDH [ˈsiːˌsaɪd]

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

Isogeny-based key exchange: High-level view

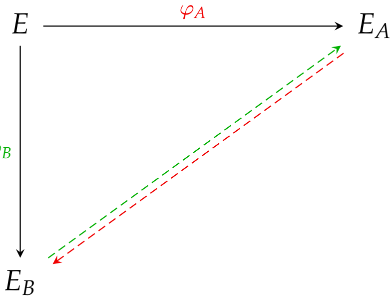
E

Isogeny-based key exchange: High-level view



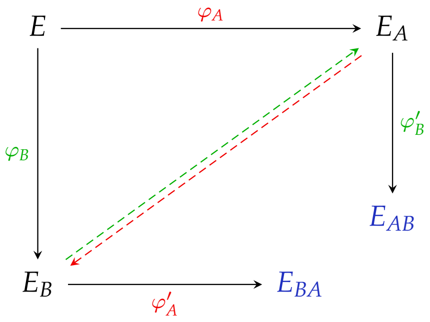
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)

Isogeny-based key exchange: High-level view



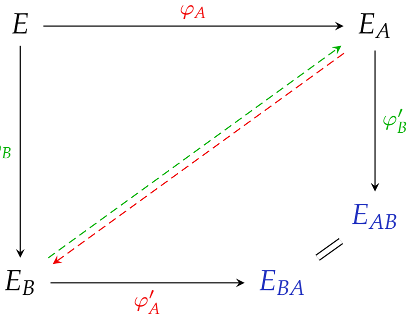
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.

Isogeny-based key exchange: High-level view



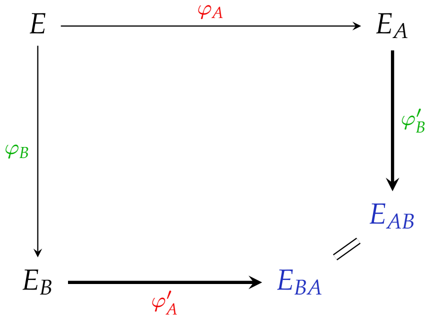
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob <u>somehow</u> finds $\varphi_{B'}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

Isogeny-based key exchange: High-level view

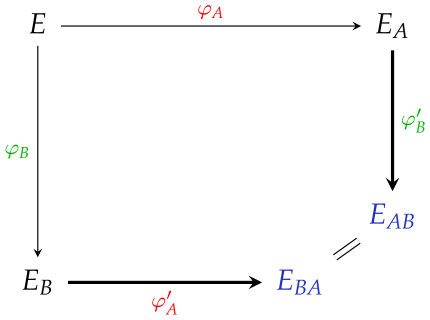


- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice <u>somehow</u> finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob <u>somehow</u> finds $\varphi_{B'}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

How to find "parallel" isogenies?



How to find “parallel” isogenies?



CSIDH’s solution (earlier: Couveignes, Rostovtsev–Stolbunov):

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 $\varphi_\ell \colon E \to E'$.

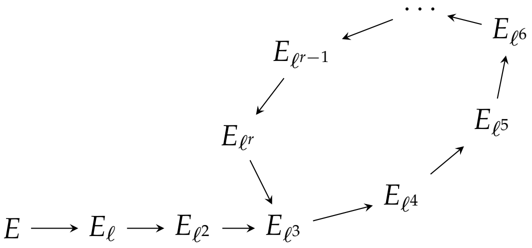
We consider prime ℓ and refer to φ_ℓ as a “special” isogeny.

Cycles from "special" isogenies

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

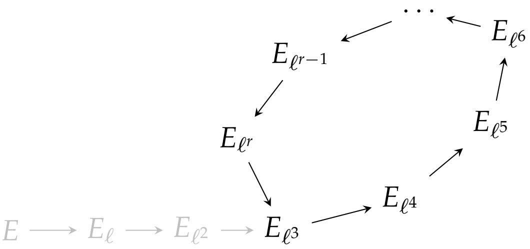
Cycles from "special" isogenies

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



Cycles from “special” isogenies

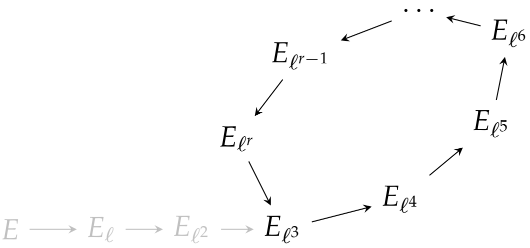
What happens when we iterate such a “special” isogeny?



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

Cycles from “special” isogenies

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

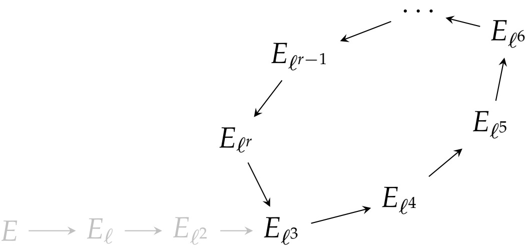


- <u>Exercise:</u> Each curve has only one other rational ℓ-isogeny.

!! 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?



- <u>Exercise:</u> Each curve has only one other rational ℓ-isogeny.

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

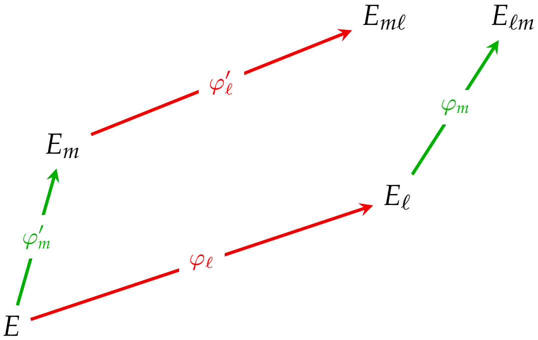
$\Longrightarrow$ The "special" isogenies φ_ℓ form isogeny cycles!

Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?

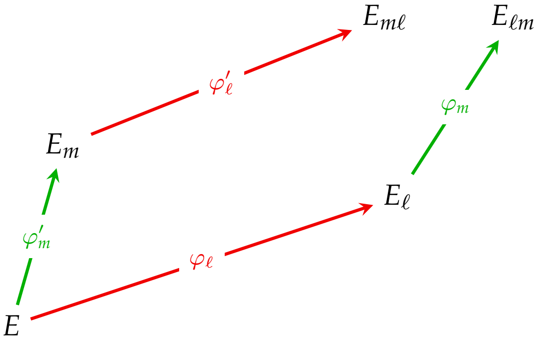
Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?



Compatible cycles from "special" isogenies

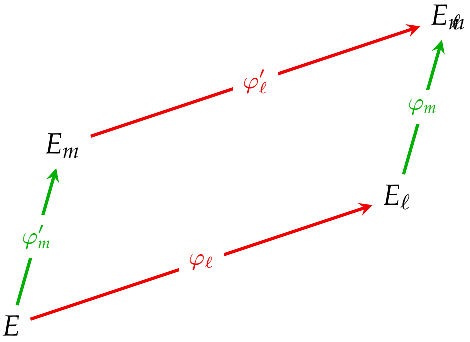
What happens when we compose those "special" isogenies?



- 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?



- 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

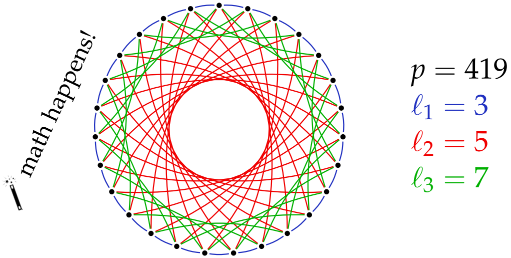
- Choose some small odd primes $\ell_1, ..., \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.

CSIDH in one slide

- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- 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.

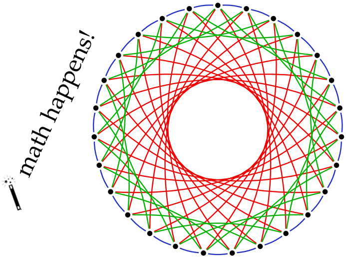
CSIDH in one slide

- 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.



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.



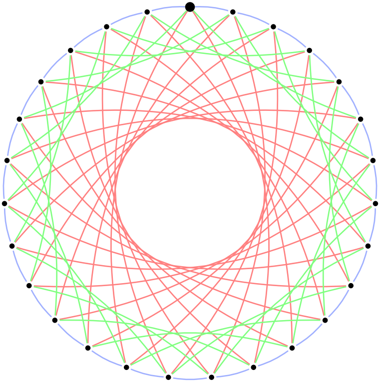
$p = 419$
$\ell_1 = 3$
$\ell_2 = 5$
$\ell_3 = 7$

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

CSIDH key exchange

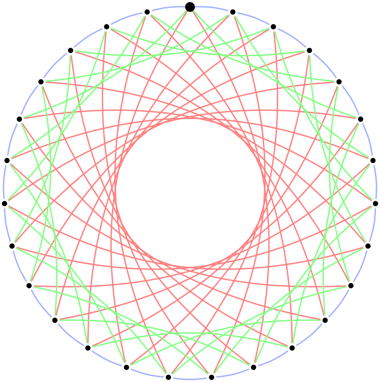
Alice

[+, +, −, −]



Bob

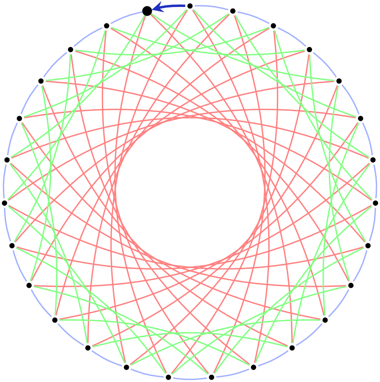
[−, +, −, −]



CSIDH key exchange

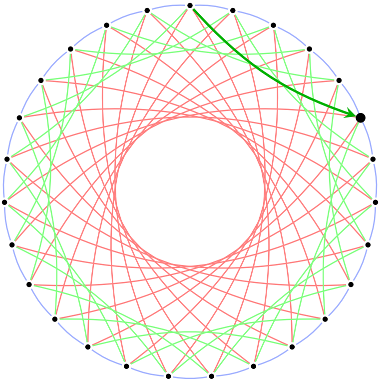
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



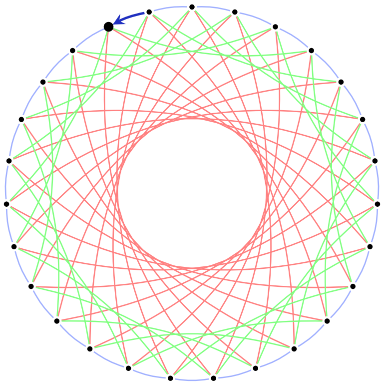
CSIDH key exchange

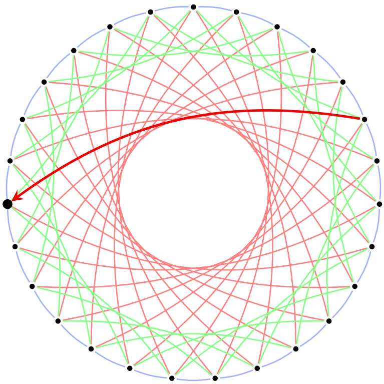
Alice

[+, +, −, −]
↑

Bob

[−, +, −, −]
↑

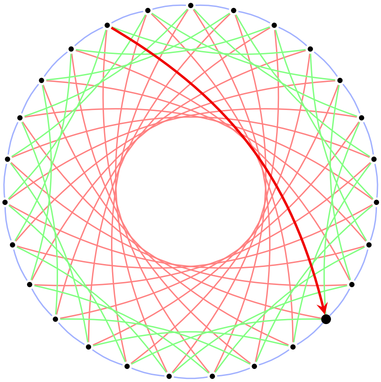




CSIDH key exchange

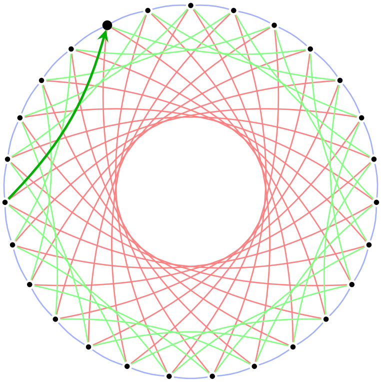
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



CSIDH key exchange

Alice

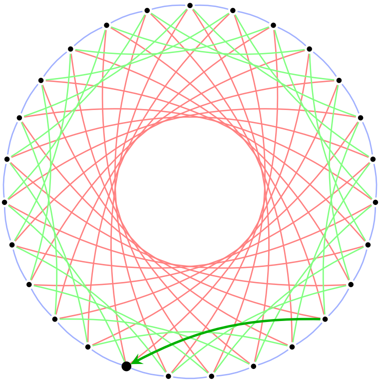
[+, +, −, −]

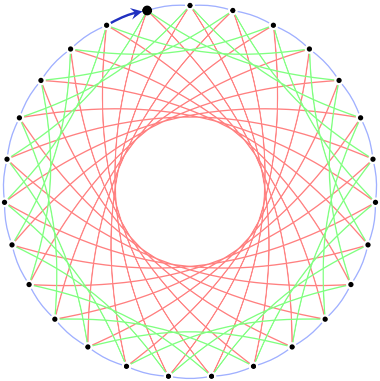
↑

Bob

[−, +, −, −]

↑





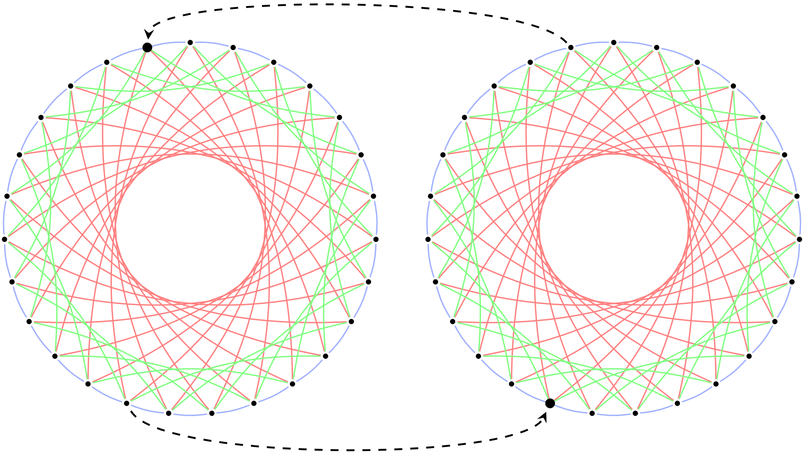
CSIDH key exchange

Alice

[+, +, −, −]

Bob

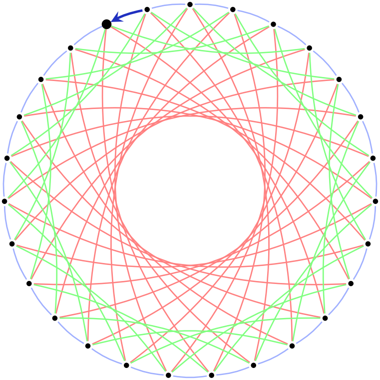
[−, +, −, −]



CSIDH key exchange

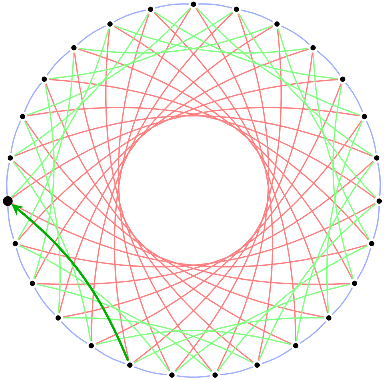
Alice

[+, +, −, −]
↑



Bob

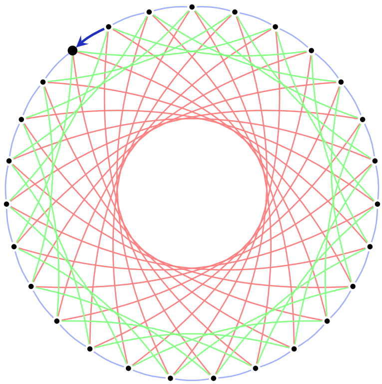
[−, +, −, −]
↑



CSIDH key exchange

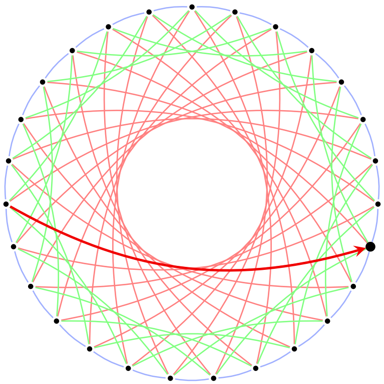
Alice

[**+**, **+**, **−**, **−**]
↑



Bob

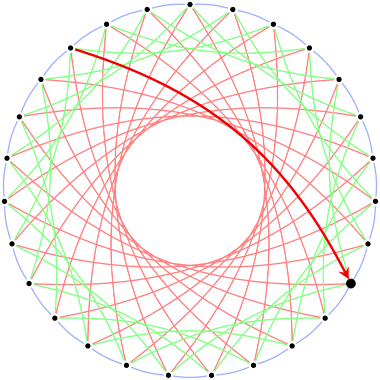
[**−**, **+**, **−**, **−**]
↑



CSIDH key exchange

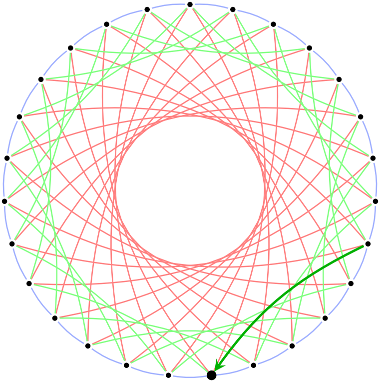
Alice

[+, +, −, −]
↑



Bob

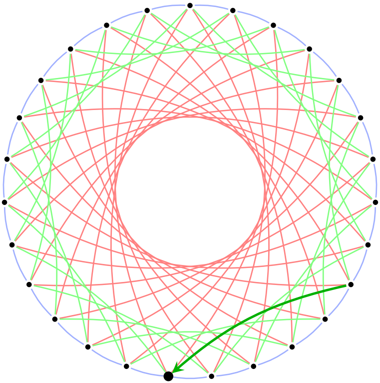
[−, +, −, −]
↑



CSIDH key exchange

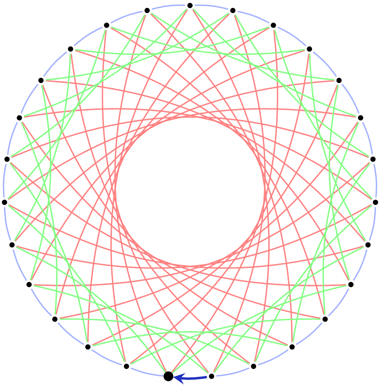
Alice

[+, +, −, −]
↑



Bob

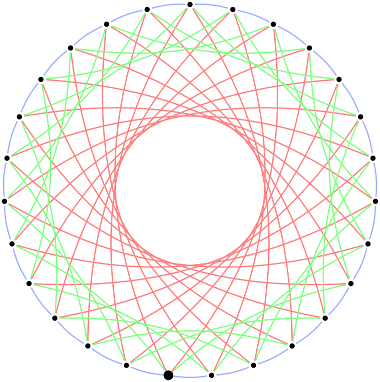
[−, +, −, −]
↑



CSIDH key exchange

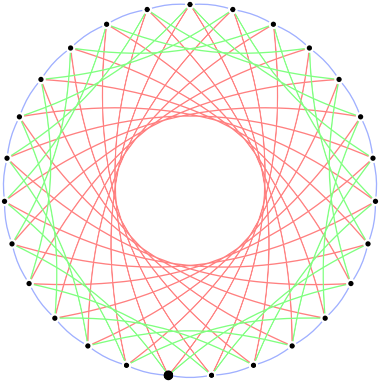
Alice

[+, +, −, −]



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

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

Recall: 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 $\underline{v} \in \mathbb{Z}^n \setminus \{0\}$ 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 $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

(I will talk some more about this later.)

The class group

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

!! The set X is **finite** $\implies$ 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 $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

(I will talk some more about this later.)

!! 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: continue
....:     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: continue
....:       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 $-P = (x, -y)$.

$\Longrightarrow$ 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 $\mathtt{xMUL}_n$ on x-coordinates such that

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

The same reasoning works for isogeny formulas.

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

Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies. ✓
- The CSIDH non-interactive key exchange. ✓
- Classical and quantum security of CSIDH.
- Orientations and the SCALLOP family.
- *Un*restricted effective group actions.

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 $b * s$!

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a path $E \to E'$ in the isogeny graph.

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a path $E \to E'$ in the isogeny graph.

The size of X is $\#\mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) = 3 \cdot h(-p) \approx \sqrt{p}$.

$\rightsquigarrow$ best known classical attack: meet-in-the-middle, $\tilde{\mathcal{O}}(p^{1/4})$.

Fully exponential: Complexity $\exp\big((\log p)^{1+o(1)}\big)$.

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a path $E \to E'$ in the isogeny graph.

The size of X is $\#\mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) = 3 \cdot h(-p) \approx \sqrt{p}$.

$\rightsquigarrow$ best known classical attack: meet-in-the-middle, $\tilde{\mathcal{O}}(p^{1/4})$.

Fully exponential: Complexity $\exp\big((\log p)^{1+o(1)}\big)$.

Solving abelian hidden shift breaks CSIDH.

$\rightsquigarrow$ non-devastating quantum attack (Kuperberg's algorithm).

Subexponential: Complexity $\exp\big((\log p)^{1/2+o(1)}\big)$.

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.

  How to compare costs?

  (Is one qubit operation $\approx$ one bit operation? a hundred? millions?)

CSIDH vs. Kuperberg

Kuperberg's algorithm consists of two components:

1. Evaluate the group action many times. ("oracle calls")
2. Combine the results in a certain way. ("sieving")

- The algorithm admits many different tradeoffs.
- Oracle calls are expensive.
- The sieving phase has classical *and* quantum operations.
  How to compare costs?
  (Is one qubit operation $\approx$ one bit operation? a hundred? millions?)

$\Longrightarrow$ Security estimates for CSIDH vary wildly.

Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies. ✓
- The CSIDH non-interactive key exchange. ✓
- Classical and quantum security of CSIDH. ✓
- Orientations and the SCALLOP family.
- *Un*restricted effective group actions.

More endomorphisms

- In CSIDH, we've used kernels of the form $K = E(\mathbb{F}_p)[\ell_i]$.

More endomorphisms

- In CSIDH, we've used kernels of the form $K = E(\mathbb{F}_p)[\ell_i]$.
- Alternative description: $K = \ker(\pi - 1) \cap \ker[\ell_i]$.
  ...so π together with scalars "carves out" our kernel subgroup!

More endomorphisms

- In CSIDH, we've used kernels of the form $K = E(\mathbb{F}_p)[\ell_i]$.
- Alternative description: $K = \ker(\pi - 1) \cap \ker[\ell_i]$.
  ...so π together with scalars "carves out" our kernel subgroup!

- New idea: Replace π by other endomorphisms.
  (Recall that $\mathrm{End}(E)$ is a rank-4 lattice in the supersingular case $\rightsquigarrow$ plenty of choice.)

More endomorphisms

- In CSIDH, we've used kernels of the form $K = E(\mathbb{F}_p)[\ell_i]$.
- Alternative description: $K = \ker(\pi - 1) \cap \ker[\ell_i]$.
  ...so π together with scalars "carves out" our kernel subgroup!
- New idea: Replace π by other endomorphisms.
  (Recall that $\mathrm{End}(E)$ is a rank-4 lattice in the supersingular case $\rightsquigarrow$ plenty of choice.)

Fact: If $\varphi\colon E \to E'$ is an isogeny for which $\ker(\varphi)$ is described in terms of scalars and some endomorphism $\tau \in \mathrm{End}(E)$, then we can usually* push τ through φ:

$$\begin{aligned} \mathbb{Z}[\tau] &\hookrightarrow \mathrm{End}(E') \\ \tau &\longmapsto (\varphi \circ \tau \circ \widehat{\varphi})/\deg(\varphi) \end{aligned}$$

* Devils in details.

Ideals $\leftrightarrow$ kernels

More precisely, the subsets of endomorphisms which determine isogeny kernel subgroups are ideals of the endomorphism ring.

Ideals $\leftrightarrow$ kernels

More precisely, the subsets of endomorphisms which determine isogeny kernel subgroups are ideals of the endomorphism ring.

Principal ideals (ϑ) correspond to endomorphisms ϑ.

Ideals $\leftrightarrow$ kernels

More precisely, the subsets of endomorphisms which determine isogeny kernel subgroups are ideals of the endomorphism ring.

Principal ideals (ϑ) correspond to endomorphisms ϑ.

$\rightsquigarrow$ Connection to the "class set" or class group:

ideals	$\longleftrightarrow$	kernels	$\longleftrightarrow$	isogenies
ideal *classes*	$\longleftrightarrow$	(no name)	$\longleftrightarrow$	isogeny *codomains*

Orientations & oriented curves

Let $\mathcal{O} = \mathbb{Z}[\tau]$ be an imaginary-quadratic order.

(Standard cases: $\tau = \sqrt{-d}$ or $\tau = \frac{1+\sqrt{-d}}{2}$ where $d \in \mathbb{Z}_{\geq 1}$.)

Orientations & oriented curves

Let $\mathcal{O} = \mathbb{Z}[\tau]$ be an imaginary-quadratic order.

(Standard cases: $\tau = \sqrt{-d}$ or $\tau = \frac{1+\sqrt{-d}}{2}$ where $d \in \mathbb{Z}_{\geq 1}$.)

An $\mathcal{O}$-orientation of an elliptic curve E is a ring embedding

$$\iota \colon \mathcal{O} \hookrightarrow \operatorname{End}(E)\,.$$

The pair (E, ι) is then called an $\mathcal{O}$-oriented curve.

Orientations & oriented curves

Let $\mathcal{O} = \mathbb{Z}[\tau]$ be an imaginary-quadratic order.

(Standard cases: $\tau = \sqrt{-d}$ or $\tau = \frac{1+\sqrt{-d}}{2}$ where $d \in \mathbb{Z}_{\geq 1}$.)

An $\mathcal{O}$-orientation of an elliptic curve E is a ring embedding

$$\iota \colon \mathcal{O} \hookrightarrow \mathrm{End}(E)\,.$$

The pair (E, ι) is then called an $\mathcal{O}$-oriented curve.

Example: For $E/\mathbb{F}_p$ supersingular with $p \geq 5$, there are two orientations by $\mathbb{Z}[\sqrt{-p}]$: Mapping $\sqrt{-p}$ either to π or to $-\pi$.

Orientations & oriented curves

Let $\mathcal{O} = \mathbb{Z}[\tau]$ be an imaginary-quadratic order.

(Standard cases: $\tau = \sqrt{-d}$ or $\tau = \frac{1+\sqrt{-d}}{2}$ where $d \in \mathbb{Z}_{\geq 1}$.)

An $\mathcal{O}$-orientation of an elliptic curve E is a ring embedding

$$\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)\,.$$

The pair (E, ι) is then called an $\mathcal{O}$-oriented curve.

Example: For $E/\mathbb{F}_p$ supersingular with $p \geq 5$, there are two orientations by $\mathbb{Z}[\sqrt{-p}]$: Mapping $\sqrt{-p}$ either to π or to $-\pi$.

Example: Any nonscalar endomorphism $\tau \in \mathrm{End}(E) \setminus \mathbb{Z}$ defines an orientation of $\mathcal{O} := \mathbb{Z}[\tau]$ on E.

The oriented class-group action

Onuki 2020 (previously Kohel–Colò without proof):

Theorem 3.4. *Let K be an imaginary quadratic field such that p does not split in K, and $\mathcal{O}$ an order in K such that p does not divide the conductor of $\mathcal{O}$. Then the ideal class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$.*

https://arxiv.org/pdf/2002.09894

The oriented class-group action

Onuki 2020 (previously Kohel–Colò without proof):

Theorem 3.4. *Let K be an imaginary quadratic field such that p does not split in K, and $\mathcal{O}$ an order in K such that p does not divide the conductor of $\mathcal{O}$. Then the ideal class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$.*

https://arxiv.org/pdf/2002.09894

$\rho(\mathcal{E}\ell\ell(\mathcal{O}))$: a set of supersingular elliptic curves E over $\mathbb{F}_{p^2}$ with a primitive orientation $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$, up to oriented isomorphism.

- $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$ is *primitive* if $(\iota(\mathcal{O}) \otimes_{\mathbb{Z}} \mathbb{Q}) \cap \mathrm{End}(E) = \iota(\mathcal{O})$.
- $\alpha\colon (E,\iota) \to (E',\iota')$ is an *oriented* isomorphism if $\alpha \circ \iota = \iota' \circ \alpha$.

The oriented class-group action

Onuki 2020 (previously Kohel–Colò without proof):

Theorem 3.4. *Let K be an imaginary quadratic field such that p does not split in K, and $\mathcal{O}$ an order in K such that p does not divide the conductor of $\mathcal{O}$. Then the ideal class group $\mathcal{C}\ell(\mathcal{O})$ acts freely and transitively on $\rho(\mathcal{E}\ell\ell(\mathcal{O}))$.*

https://arxiv.org/pdf/2002.09894

$\rho(\mathcal{E}\ell\ell(\mathcal{O}))$: a set of supersingular elliptic curves E over $\mathbb{F}_{p^2}$ with a primitive orientation $\iota : \mathcal{O} \hookrightarrow \mathrm{End}(E)$, up to oriented isomorphism.

- $\iota : \mathcal{O} \hookrightarrow \mathrm{End}(E)$ is *primitive* if $(\iota(\mathcal{O}) \otimes_{\mathbb{Z}} \mathbb{Q}) \cap \mathrm{End}(E) = \iota(\mathcal{O})$.
- $\alpha : (E, \iota) \to (E', \iota')$ is an *oriented* isomorphism if $\alpha \circ \iota = \iota' \circ \alpha$.

The group action is defined as follows:

$$\mathfrak{a} \star (E, \iota) := \left(E/\mathfrak{a},\ (\phi_{\mathfrak{a}} \circ \iota \circ \widehat{\phi}_{\mathfrak{a}})/\mathrm{norm}(\mathfrak{a})\right)$$

where $\phi_{\mathfrak{a}} : E \to E/\mathfrak{a}$ is the isogeny with kernel

$$E[\mathfrak{a}] := \bigcap_{\alpha \in \mathfrak{a}} \ker(\iota(\alpha)) .$$

Recap: CSIDH

Recall that in CSIDH, our isogeny kernels are generated by $(x, y) \in E$ with $x \in \mathbb{F}_p$.

Recap: CSIDH

Recall that in CSIDH, our isogeny kernels are generated by $(x, y) \in E$ with $x \in \mathbb{F}_p$. The two cases $y \in \mathbb{F}_p$ and $y \notin \mathbb{F}_p$ correspond precisely to the two $\mathbb{Z}[\pi]$-ideals

$$\mathfrak{l}_i := (\ell_i, \pi - 1)\,;$$
$$\overline{\mathfrak{l}_i} := (\ell_i, \pi + 1)\,.$$

where π is the p-power Frobenius endomorphism ($\pi^2 = [-p]$).

Recap: CSIDH

Recall that in CSIDH, our isogeny kernels are generated by $(x, y) \in E$ with $x \in \mathbb{F}_p$. The two cases $y \in \mathbb{F}_p$ and $y \notin \mathbb{F}_p$ correspond precisely to the two $\mathbb{Z}[\pi]$-ideals

$$\mathfrak{l}_i := (\ell_i, \pi - 1)\,;$$
$$\overline{\mathfrak{l}_i} := (\ell_i, \pi + 1)\,.$$

where π is the p-power Frobenius endomorphism ($\pi^2 = [-p]$).

Since "finding" π on any $E/\mathbb{F}_p$ is trivial (it is $\pi\colon (x, y) \mapsto (x^p, y^p)$), it need not be transmitted and we get an action on curves only.

Recap: CSIDH

Recall that in CSIDH, our isogeny kernels are generated by $(x, y) \in E$ with $x \in \mathbb{F}_p$. The two cases $y \in \mathbb{F}_p$ and $y \notin \mathbb{F}_p$ correspond precisely to the two $\mathbb{Z}[\pi]$-ideals

$$\mathfrak{l}_i := (\ell_i, \pi - 1)\,;$$
$$\overline{\mathfrak{l}_i} := (\ell_i, \pi + 1)\,.$$

where π is the p-power Frobenius endomorphism ($\pi^2 = [-p]$).

Since "finding" π on any $E/\mathbb{F}_p$ is trivial (it is $\pi\colon (x, y) \mapsto (x^p, y^p)$), it need not be transmitted and we get an action on curves only.

Fun fact: Orienting $E/\mathbb{F}_p$ by $\sqrt{-p} \mapsto -\pi$ gives exactly the same picture, but everything is mirrored along "quadratic twisting":

$$\{y^2 = x^3 + Ax^2 + x\} \xrightarrow{\sim} \{y^2 = x^3 - Ax^2 + x\}$$

Representing orientations

To turn the previous theorem into a concrete group action for general $\mathcal{O}$, we need to specify how to encode the pair (E, ι):

Representing orientations

To turn the previous theorem into a concrete group action for general $\mathcal{O}$, we need to specify how to encode the pair (E, ι):

- When $\mathcal{O}$ is represented as $\mathbb{Z}[\tau] := \mathbb{Z}[X]/\mu_\tau(X)$ where μ_τ is the minimal polynomial of τ, an embedding $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$ can be specified by the image $\iota(\tau)$.

Representing orientations

To turn the previous theorem into a concrete group action for general $\mathcal{O}$, we need to specify how to encode the pair (E, ι):

- When $\mathcal{O}$ is represented as $\mathbb{Z}[\tau] := \mathbb{Z}[X]/\mu_\tau(X)$ where μ_τ is the minimal polynomial of τ, an embedding $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$ can be specified by the image $\iota(\tau)$.

$\rightsquigarrow$ In practice, an oriented curve is given as a pair (E, ϑ) with $\vartheta \in \mathrm{End}(E)$, implicitly communicating that $\vartheta = \iota(\tau)$.

Representing orientations

To turn the previous theorem into a concrete group action for general $\mathcal{O}$, we need to specify how to encode the pair (E, ι):

- When $\mathcal{O}$ is represented as $\mathbb{Z}[\tau] := \mathbb{Z}[X]/\mu_\tau(X)$ where μ_τ is the minimal polynomial of τ, an embedding $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$ can be specified by the image $\iota(\tau)$.

$\rightsquigarrow$ In practice, an oriented curve is given as a pair (E, ϑ) with $\vartheta \in \mathrm{End}(E)$, implicitly communicating that $\vartheta = \iota(\tau)$.

- There are multiple options for representing such a ϑ.
  Simple example: A deterministically chosen generator point of $\ker(\vartheta)$.
  More complicated: Deterministic "HD" representation (SCALLOP-HD).

Oriented isogeny group actions: Why?

- Key point: Orientations allow us to decouple the discriminant of $\mathcal{O}$ from the characteristic p.

This is advantageous for at least two reasons (see next part):

Oriented isogeny group actions: Why?

- Key point: Orientations allow us to decouple the discriminant of $\mathcal{O}$ from the characteristic p.

This is advantageous for at least two reasons (see next part):

$\rightsquigarrow$ Can use rings like $\mathcal{O} = \mathbb{Z}[\sqrt{-f^2 d}]$, where computing the relation lattice Λ can be much easier than for general $\mathcal{O}$.

Oriented isogeny group actions: Why?

- Key point: Orientations allow us to decouple the discriminant of $\mathcal{O}$ from the characteristic p.

This is advantageous for at least two reasons (see next part):

$\rightsquigarrow$ Can use rings like $\mathcal{O} = \mathbb{Z}[\sqrt{-f^2 d}]$, where computing the relation lattice Λ can be much easier than for general $\mathcal{O}$.

$\rightsquigarrow$ For Clapoti, we have to solve norm equations that are derived from $\mathcal{O}$ for target values derived from p.

Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies. ✓
- The CSIDH non-interactive key exchange. ✓
- Classical and quantum security of CSIDH. ✓
- Orientations and the SCALLOP family. ✓
- *Un*restricted effective group actions.

The basic strategy à la C/R–S

- Let $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ be small prime ideals of $\mathcal{O}$, and suppose $\mathfrak{a}$ is given to us in the form $\mathfrak{a} = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$.
- Then $\mathfrak{a}$ can be evaluated as a sequence of $\mathfrak{l}_i$.

The basic strategy à la C/R–S

- Let $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ be small prime ideals of $\mathcal{O}$, and <u>suppose</u> $\mathfrak{a}$ is given to us in the form $\mathfrak{a} = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$.
- Then $\mathfrak{a}$ can be evaluated as a sequence of $\mathfrak{l}_i$.

- Evaluating a single $\mathfrak{l}_i$: Write $\mathfrak{l}_i = (\ell_i, \vartheta - \lambda_i)$. Then the kernel is an order-ℓ_i point P with $\vartheta(P) = [\lambda_i]P$.

The basic strategy à la C/R–S

- Let $\mathfrak{l}_1, ..., \mathfrak{l}_n$ be small prime ideals of $\mathcal{O}$, and <u>suppose</u> $\mathfrak{a}$ is given to us in the form $\mathfrak{a} = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$.
- Then $\mathfrak{a}$ can be evaluated as a sequence of $\mathfrak{l}_i$.

- Evaluating a single $\mathfrak{l}_i$: Write $\mathfrak{l}_i = (\ell_i, \vartheta - \lambda_i)$. Then the kernel is an order-ℓ_i point P with $\vartheta(P) = [\lambda_i]P$.
- Optimizations: Batch multiple $\mathfrak{l}_i$ together $\rightsquigarrow$ "strategies".

The basic problem with the basic strategy

- Couveignes: This gives a “hard homogeneous space” (weirder name for a one-way commutative group action).
- The CSIDH paper repeats this.

The basic problem with the basic strategy

- Couveignes: This gives a "hard homogeneous space" (weirder name for a one-way commutative group action).
- The CSIDH paper repeats this.

<u>Issue:</u>

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  - $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.

The basic problem with the basic strategy

- Couveignes: This gives a "hard homogeneous space" (weirder name for a one-way commutative group action).
- The CSIDH paper repeats this.

Issue:

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  - $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.
- Representing $\mathrm{cl}(\mathcal{O})$ as reduced ideals allows computing in $\mathrm{cl}(\mathcal{O})$ efficiently, but evaluation becomes superpolynomial.

(A similar approach will be discussed on the following slides.)

The basic problem with the basic strategy

- Couveignes: This gives a "hard homogeneous space" (weirder name for a one-way commutative group action).
- The CSIDH paper repeats this.

Issue:

- Representing $\mathrm{cl}(\mathcal{O})$ by the group $(\mathbb{Z}^n, +)$ of exponents makes the exponents grow larger with each operation.
  $\rightsquigarrow$ Cost of evaluating after k operations is $O(\exp(k))$.
- Representing $\mathrm{cl}(\mathcal{O})$ as reduced ideals allows computing in $\mathrm{cl}(\mathcal{O})$ efficiently, but evaluation becomes superpolynomial.
  (A similar approach will be discussed on the following slides.)

$\rightsquigarrow$ A priori **not** an effective group action when done either way!

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. "Computing the class group": Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. "Lattice reduction": Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. "Approximate CVP": Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. "Isogeny steps": Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. "Computing the class group": Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. "Lattice reduction": Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. "Approximate CVP": Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. "Isogeny steps": Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh paper (2019) does all this in practice for 512-bit p.

The CSI-FiSh approach

...combines exponent vectors with reduction by exploiting the relation lattice of the chosen ideal classes. It works as follows:

The strategy to act by a given, arbitrarily long and ugly exponent vector $\underline{v} \in \mathbb{Z}^d$ consists of the following steps:

1. <u>"Computing the class group"</u>: Find a basis of the *relation lattice* $\Lambda \subseteq \mathbb{Z}^d$ with respect to $\mathfrak{l}_1, \ldots, \mathfrak{l}_d$.
   [Classically subexponential-time, quantumly polynomial-time. Precomputation.]
2. <u>"Lattice reduction"</u>: Prepare a "good" basis of Λ using a lattice-reduction algorithm such as BKZ.
   [Configurable complexity-quality tradeoff by varying the block size. Precomputation.]
3. <u>"Approximate CVP"</u>: Obtain a vector $\underline{w} \in \Lambda$ such that $\|\underline{v} - \underline{w}\|_1$ is "small", using the reduced basis.
   [Polynomial-time, but the quality depends on the quality of step 2.]
4. <u>"Isogeny steps"</u>: Evaluate the action of the vector $\underline{v} - \underline{w} \in \mathbb{Z}^d$ as a sequence of $\mathfrak{l}_i$-steps.
   [Complexity depends entirely on the output quality of step 3.]

https://yx7.cc/blah/2023-04-14.html

The CSI-FiSh paper (2019) does all this in practice for 512-bit p.

What about asymptotics?

Tradeoff: Lattice part vs. isogeny part

- By increasing the number n of ideals $\mathfrak{l}_i$, we can trade off some "isogeny effort" for "lattice effort".

$\rightsquigarrow$ Sweet spot: Minimize total cost.

Tradeoff: Lattice part vs. isogeny part

- By increasing the number n of ideals $\mathfrak{l}_i$, we can trade off some "isogeny effort" for "lattice effort".

$\rightsquigarrow$ Sweet spot: Minimize total cost.

CSI-FiSh really isn't polynomial-time

It is fairly well-known that CSIDH[1] in its basic form is merely a *restricted* effective group action $G \times X \to X$: There is a small number of group elements $\mathfrak{l}_1, \ldots, \mathfrak{l}_d \in G$ whose action can be applied to arbitrary elements of X efficiently, but applying other elements (say, large products $\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_d^{e_d}$ of the $\mathfrak{l}_i$) quickly becomes infeasible as the exponents grow.

The only known method to circumvent this issue consists of a folklore strategy first employed in practice by the signature scheme CSI-FiSh. The core of the technique is to rewrite any given group element as a *short* product combination of the $\mathfrak{l}_i$, whose action can then be computed in the usual way much more affordably. (Notice how this is philosophically similar to the role of the square-and-multiply algorithm in discrete-logarithm land!)

The main point of this post is to remark that this approach is **not asymptotically efficient**, even when a quantum computer can be used, contradicting a false belief that appears to be rather common among isogeny aficionados.

$\rightsquigarrow$

- Classically: Evaluation $L_p[1/2]$. Attack $L_p[1]$.
- Quantumly: Evaluation $L_p[1/3]$. Attack $L_p[1/2]$.

https://yx7.cc/blah/2023-04-14.html

Clapoti

Even more maritime isogenies??

Noun [edit]

clapotis *m* (*plural* **clapotis**)

1. lapping of water against a surface [synonyms ▲]

Clapoti

Even more maritime isogenies??

Noun [edit]

clapotis *m* (*plural* **clapotis**)

1. lapping of water against a surface [synonyms ▲]

- Page–Robert: A polynomial-time algorithm to evaluate the isogeny group action on arbitrary ideals.

Polynomial-time group action: Clapoti

<u>Idea:</u>

- Find two ideals $\mathfrak{b}$, $\mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.

Polynomial-time group action: Clapoti

Idea:

- Find two ideals $\mathfrak{b}, \mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.

$$\begin{array}{ccc} E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E \end{array}$$

Polynomial-time group action: Clapoti

Idea:

- Find two ideals $\mathfrak{b}, \mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.

$$
\begin{array}{ccc}
E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\
{\scriptstyle \phi_{\overline{\mathfrak{c}}}}\big\downarrow & & \big\downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\
E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E
\end{array}
$$

- Kani: This gives an N-isogeny

$$
\begin{aligned}
\Phi \colon E \times E &\longrightarrow E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}, \\
(P, Q) &\longmapsto \left(\phi_{\mathfrak{b}}(P) + \widehat{\psi_{\overline{\mathfrak{c}}}}(Q),\ -\phi_{\overline{\mathfrak{c}}}(P) + \widehat{\psi_{\mathfrak{b}}}(Q)\right).
\end{aligned}
$$

Polynomial-time group action: Clapoti

<u>Idea:</u>

- Find two ideals $\mathfrak{b}, \mathfrak{c}$ of coprime norms, both equivalent to $\mathfrak{a}$.
  Let $N := \mathrm{norm}(\mathfrak{b}) + \mathrm{norm}(\mathfrak{c})$.

$$\begin{array}{ccc} E & \xrightarrow{\phi_{\mathfrak{b}}} & E_{\mathfrak{a}} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \psi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\psi_{\mathfrak{b}}]{} & E \end{array}$$

- Kani: This gives an N-isogeny

$$\begin{aligned} \Phi\colon E \times E &\longrightarrow E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}, \\ (P, Q) &\longmapsto \left(\phi_{\mathfrak{b}}(P) + \widehat{\psi}_{\overline{\mathfrak{c}}}(Q),\ -\phi_{\overline{\mathfrak{c}}}(P) + \widehat{\psi}_{\mathfrak{b}}(Q)\right). \end{aligned}$$

- The kernel is $\ker(\Phi) = \left\{\left(\widehat{\phi}_{\mathfrak{b}}(R), \psi_{\overline{\mathfrak{c}}}(R)\right) : R \in E_{\mathfrak{a}}[N]\right\}$.

Polynomial-time group action: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\bar{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.

Polynomial-time group action: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\bar{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\bar{\mathfrak{c}}}$.

Polynomial-time group action: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generator of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

Polynomial-time group action: Clapoti

- The kernel is $\ker(\Phi) = \left\{ \left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R) \right) : R \in E_{\mathfrak{a}}[N] \right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{ \left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R) \right) \mid R \in E[N] \right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generator of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

$\Longrightarrow$ The isogeny group action can now be computed in polynomial time even for "ugly" input ideals.

Polynomial-time group action: Clapoti

- The kernel is $\ker(\Phi) = \left\{\left(\widehat{\phi_{\mathfrak{b}}}(R), \psi_{\overline{\mathfrak{c}}}(R)\right) : R \in E_{\mathfrak{a}}[N]\right\}$.
- Issue: Evaluating this formula seems to require a-priori knowledge of $\phi_{\mathfrak{b}}, \psi_{\overline{\mathfrak{c}}}$.

The kernel is equal to the alternative description

$$\ker(\Phi) = \left\{\left([\mathrm{norm}(\mathfrak{b})]R, \gamma(R)\right) \mid R \in E[N]\right\}$$

where $\gamma \in \mathrm{End}(E)$ is a generator of the principal ideal $\mathfrak{b}\overline{\mathfrak{c}}$.

$\Longrightarrow$ The isogeny group action can now be computed in polynomial time even for "ugly" input ideals.

$\Longrightarrow$ Isogenies yield true effective group actions, at last!

Plan for this lecture

- High-level overview for intuition. ✓
- Recap: Elliptic curves & isogenies. ✓
- The CSIDH non-interactive key exchange. ✓
- Classical and quantum security of CSIDH. ✓
- Orientations and the SCALLOP family. ✓
- *Un*restricted effective group actions. ✓

Questions?

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