Introduction to isogeny-based cryptography

Lorenz Panny

Technische Universität München

PQCSA summer school “PQC fundamentals”,
Albena, 17 June 2025

Big picture 🔎🔎

- Isogenies are a type of maps between elliptic curves.

Big picture 🔎🔎

- Isogenies are a type of maps between elliptic curves.
- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

Big picture 🔍🔍

- Isogenies are a type of maps between elliptic curves.
- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

$\rightsquigarrow$ ***Cryptography!***

Big picture 🔎🔎

- Isogenies are a type of maps between elliptic curves.
- Sampling an isogeny *from* some curve is easy, recovering an isogeny *between* given curves seems very hard.

$\rightsquigarrow$ ***Cryptography!***

[Modern isogeny-based cryptography uses not just elliptic curves, but also higher-dimensional abelian varieties. $\rightsquigarrow$ *Friday.* ☺]

Plan for this talk

- Some high-level intuition.
- Elliptic curves & isogenies.
- Isogeny group actions.
- Signatures from isogenies.

Diffie–Hellman key exchange (1976)

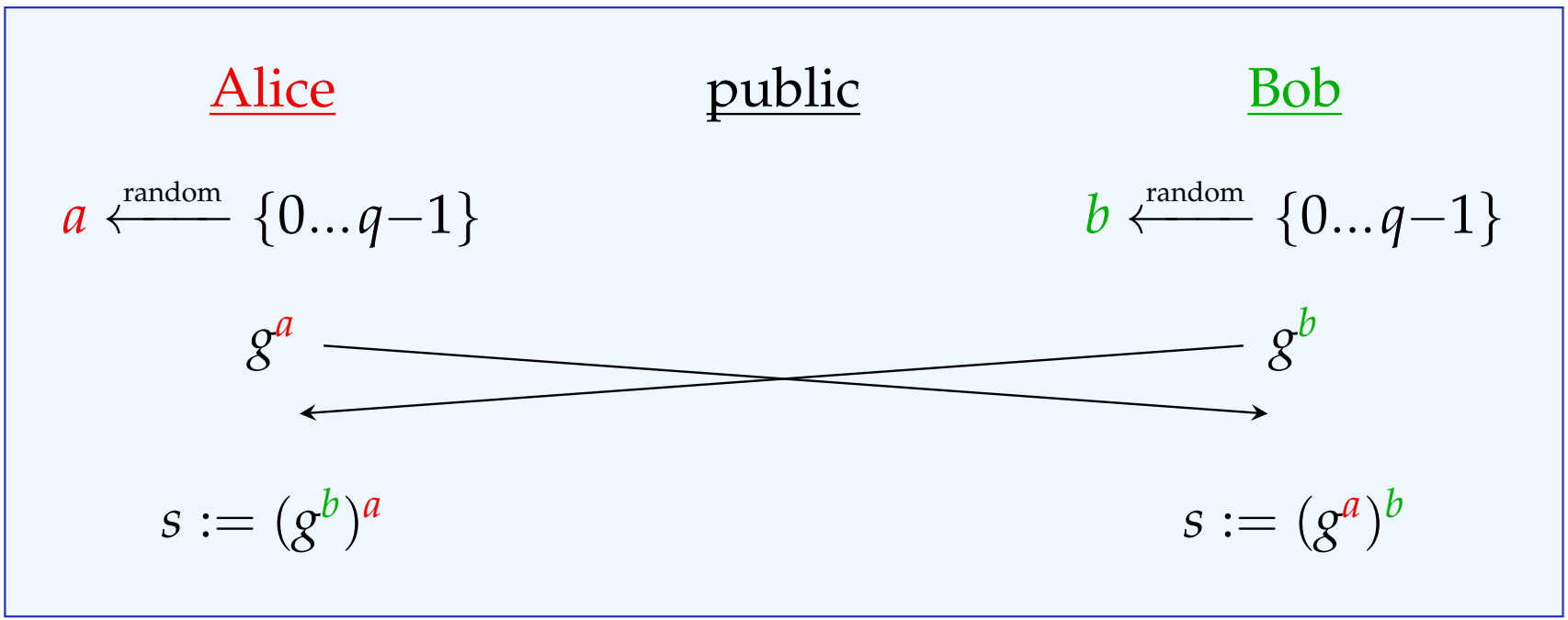
Public parameters:

- 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:

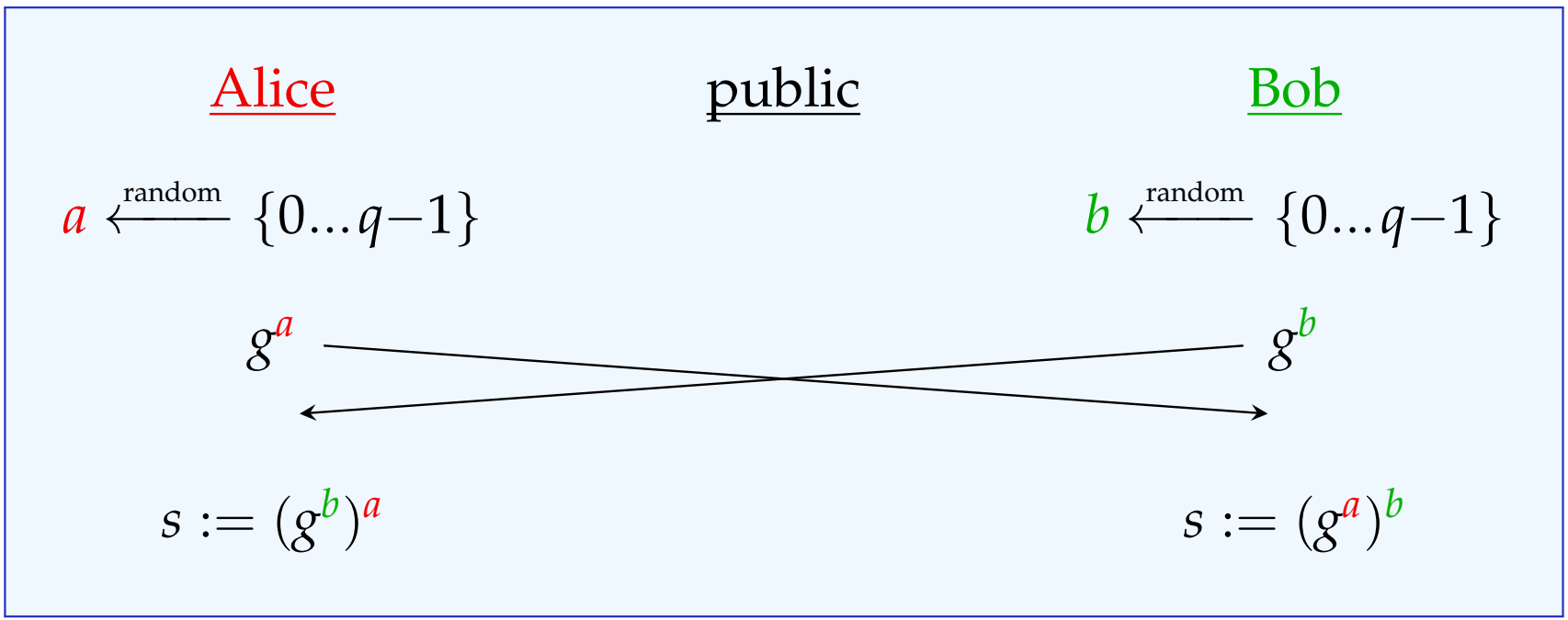
- a finite group G (traditionally $\mathbb{F}_p^*$, today elliptic curves)
- an element $g \in G$ of prime order q



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

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

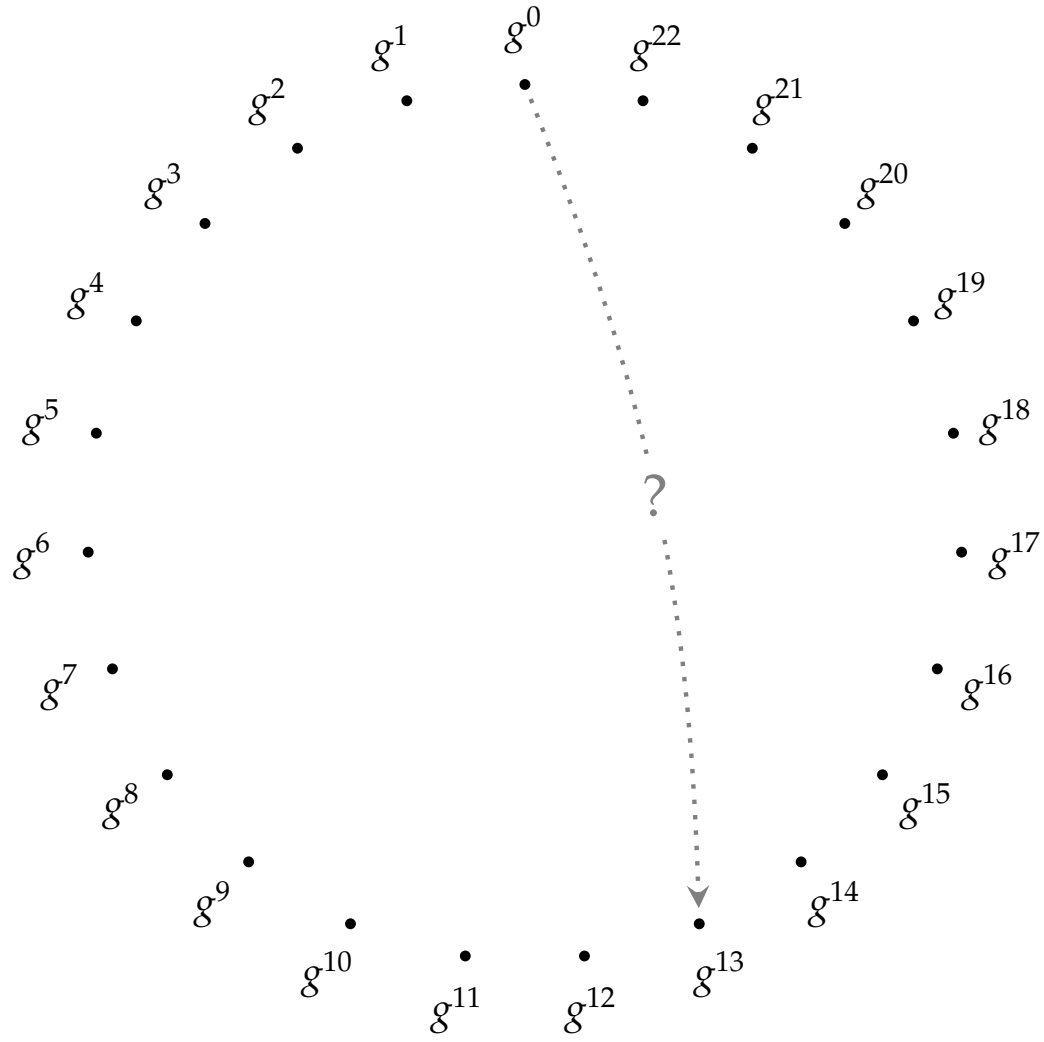
Bob	**Attacker** Eve
1. Set $t \leftarrow g$.	1. Set $t \leftarrow g$. If $t = B$ return 1.
2. Set $t \leftarrow t \cdot g$.	2. Set $t \leftarrow t \cdot g$. If $t = B$ return 2.
3. Set $t \leftarrow t \cdot g$.	3. Set $t \leftarrow t \cdot g$. If $t = B$ return 3.
4. Set $t \leftarrow t \cdot g$.	4.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

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 $b+2$.
	...

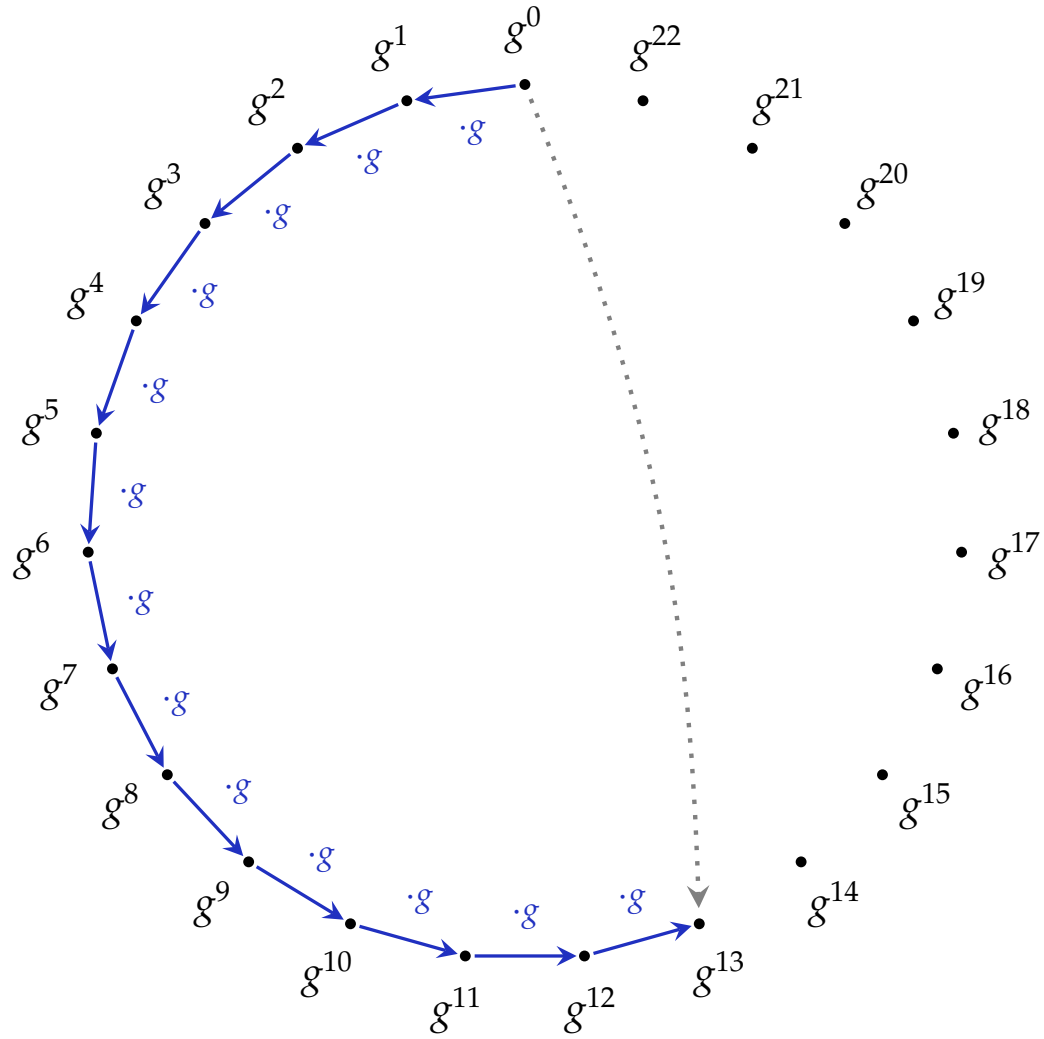
Effort for both: $O(\#G)$. Bob needs to be smarter.

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



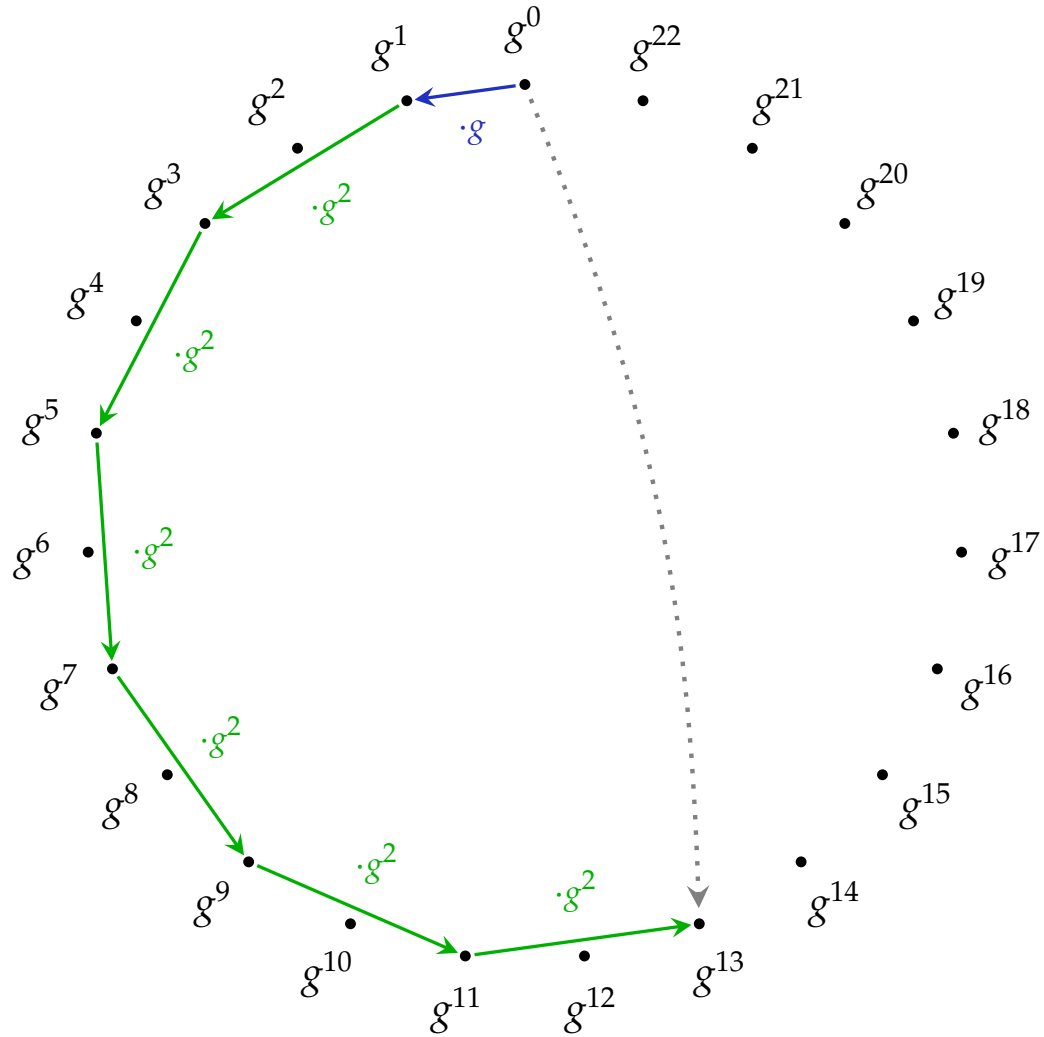
Bob computes his public key g^{13} from g.

multiply



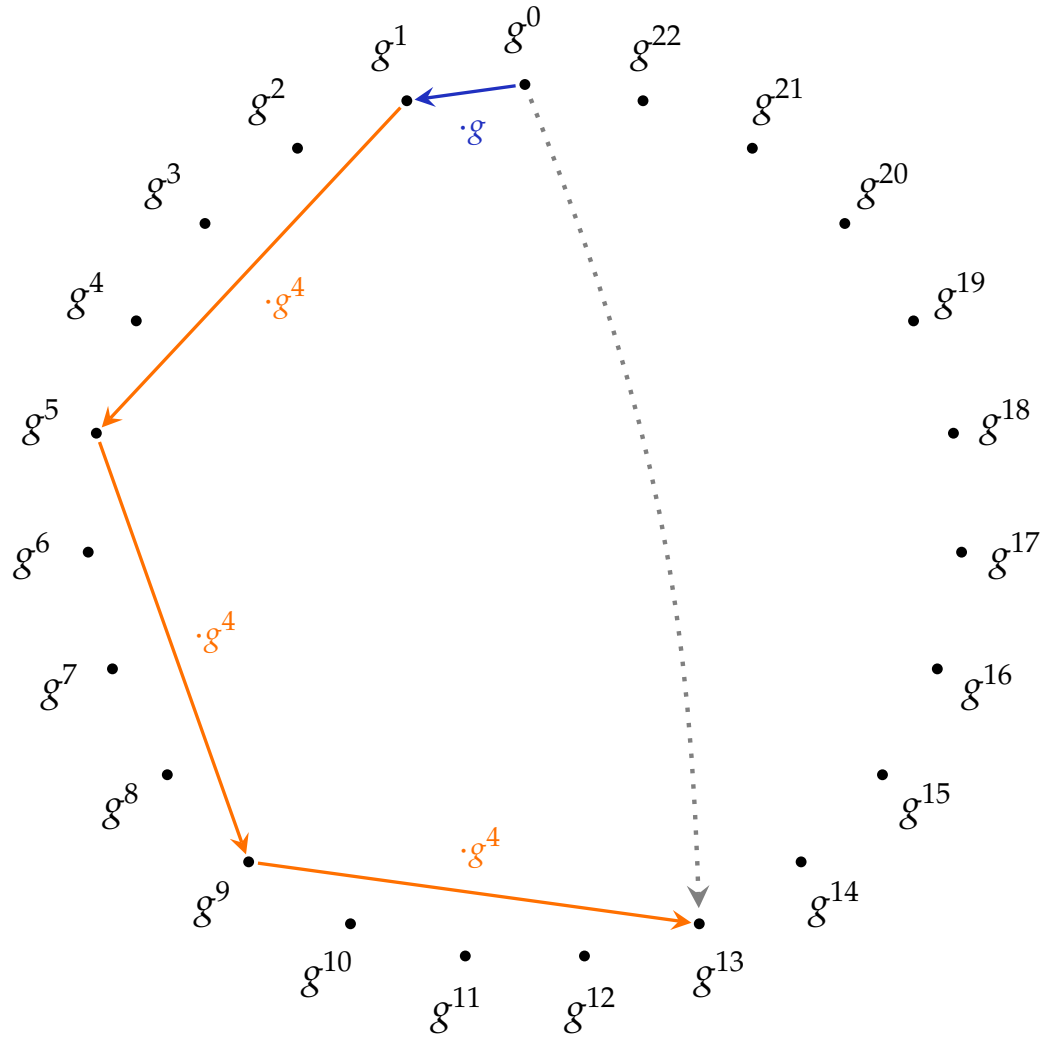
Bob computes his public key g^{13} from g.

Square-and-multiply



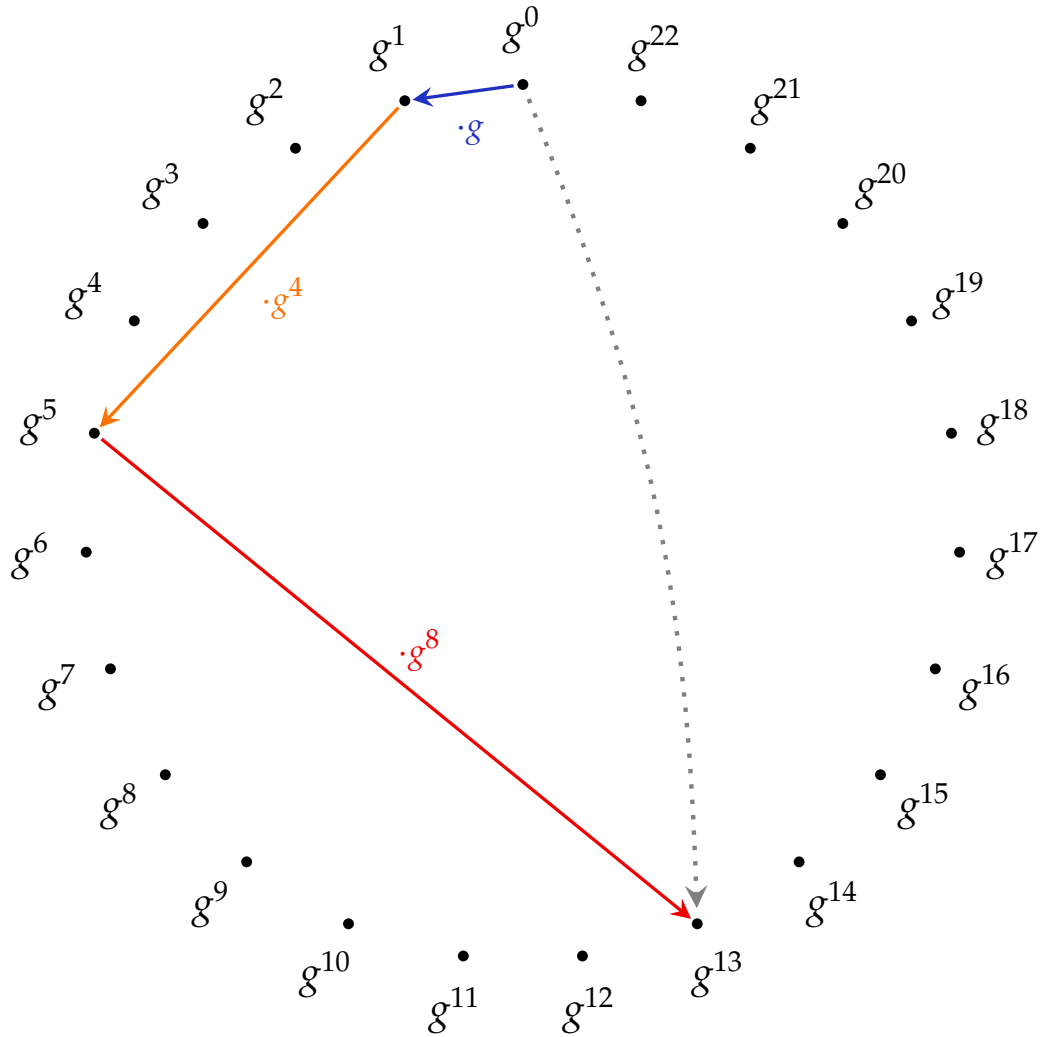
Bob computes his public key g^{13} from g.

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



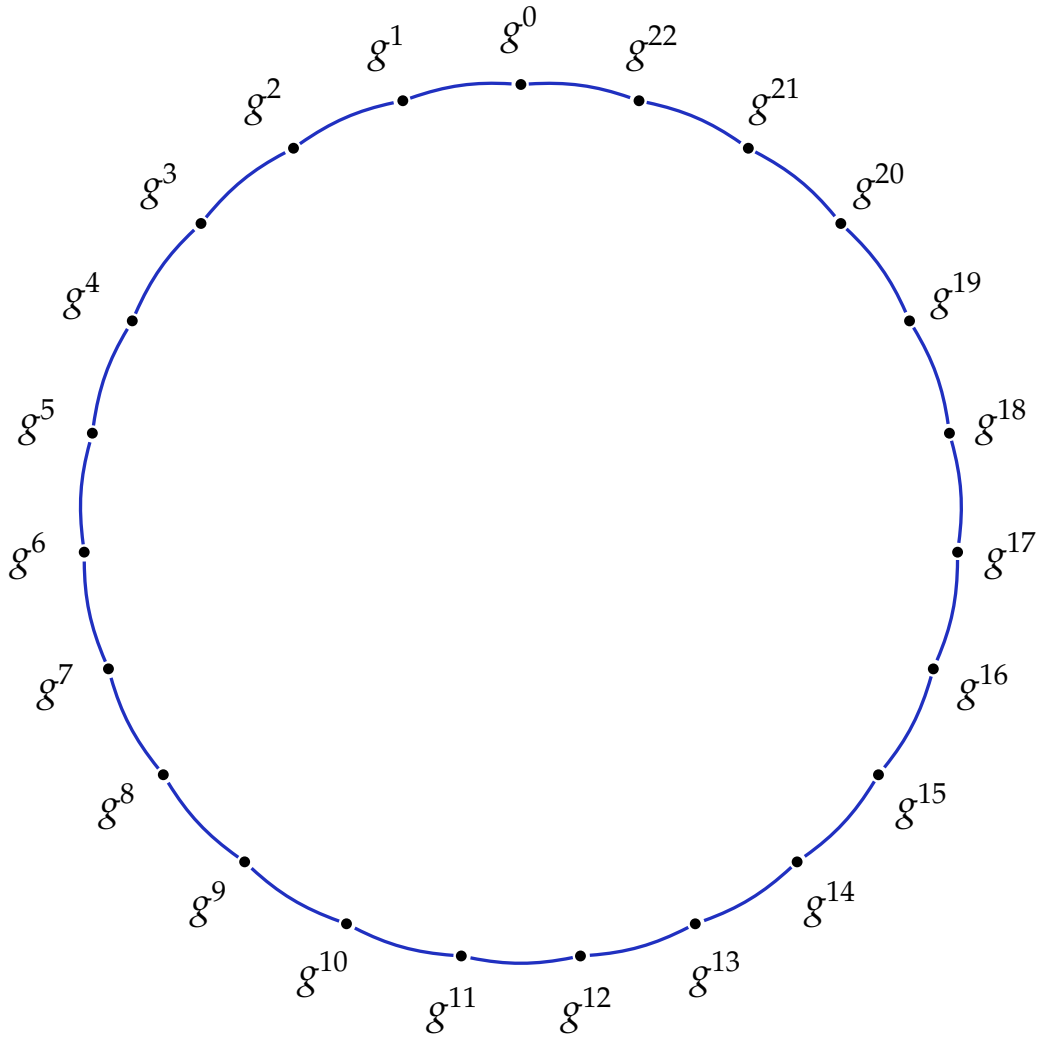
Bob computes his public key g^{13} from g.

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

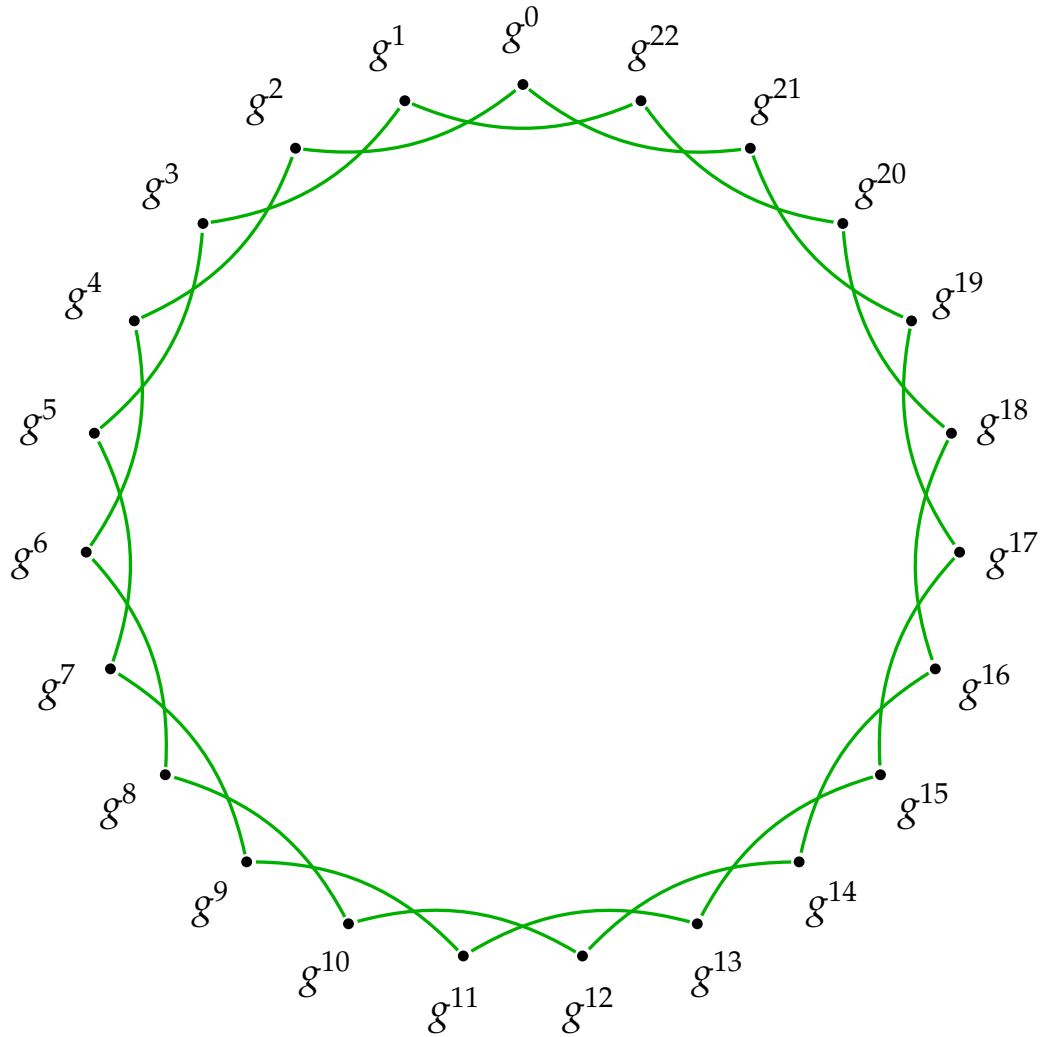


Bob computes his public key g^{13} from g.

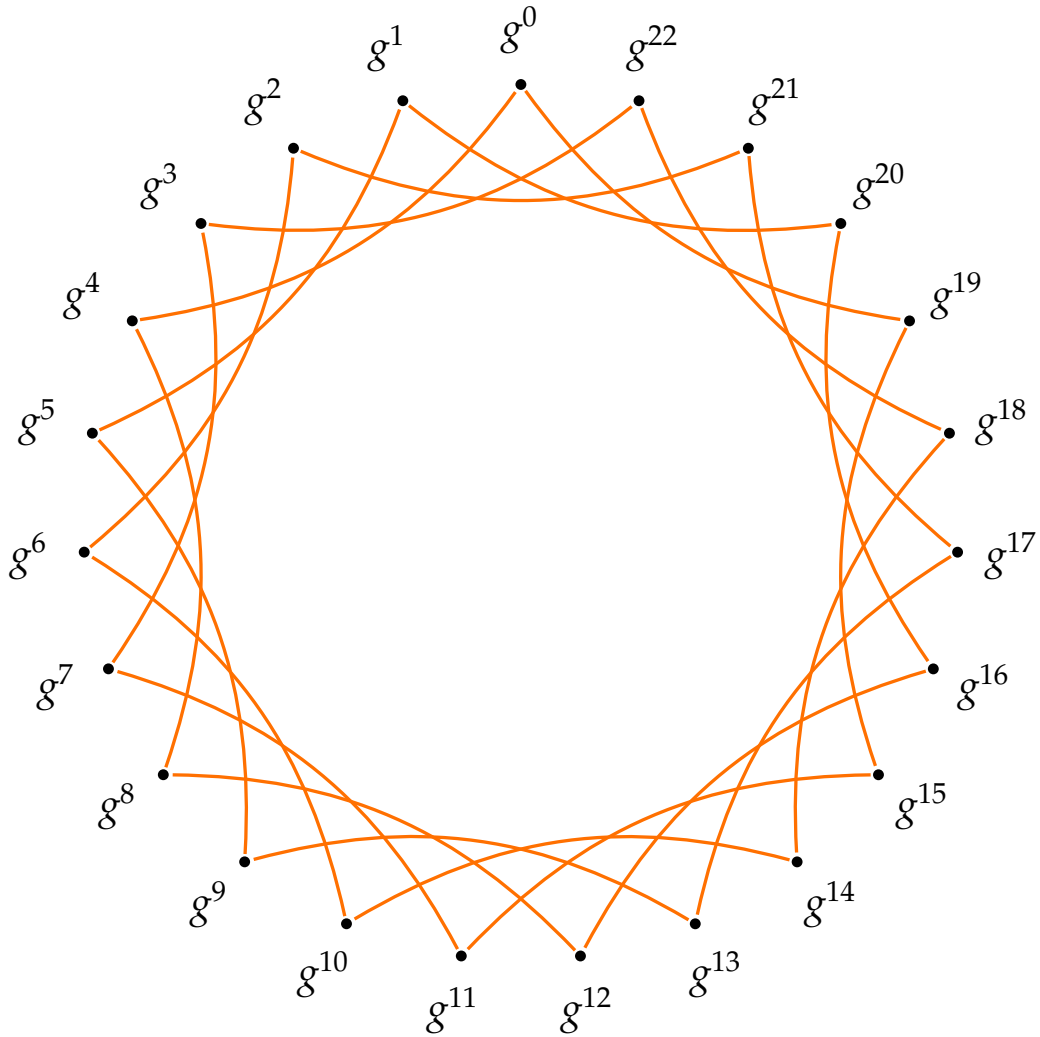
Square-and-multiply as graphs



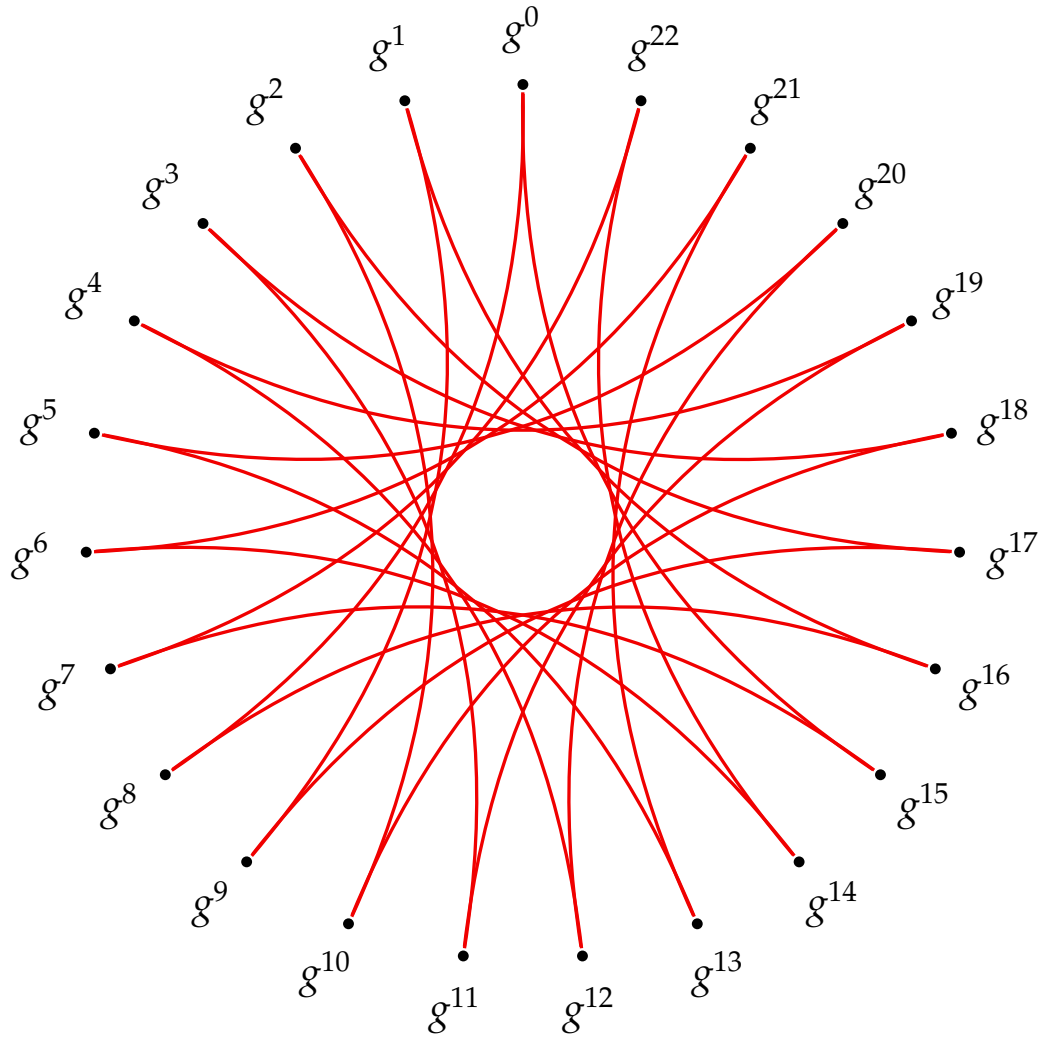
Square-and-multiply as graphs



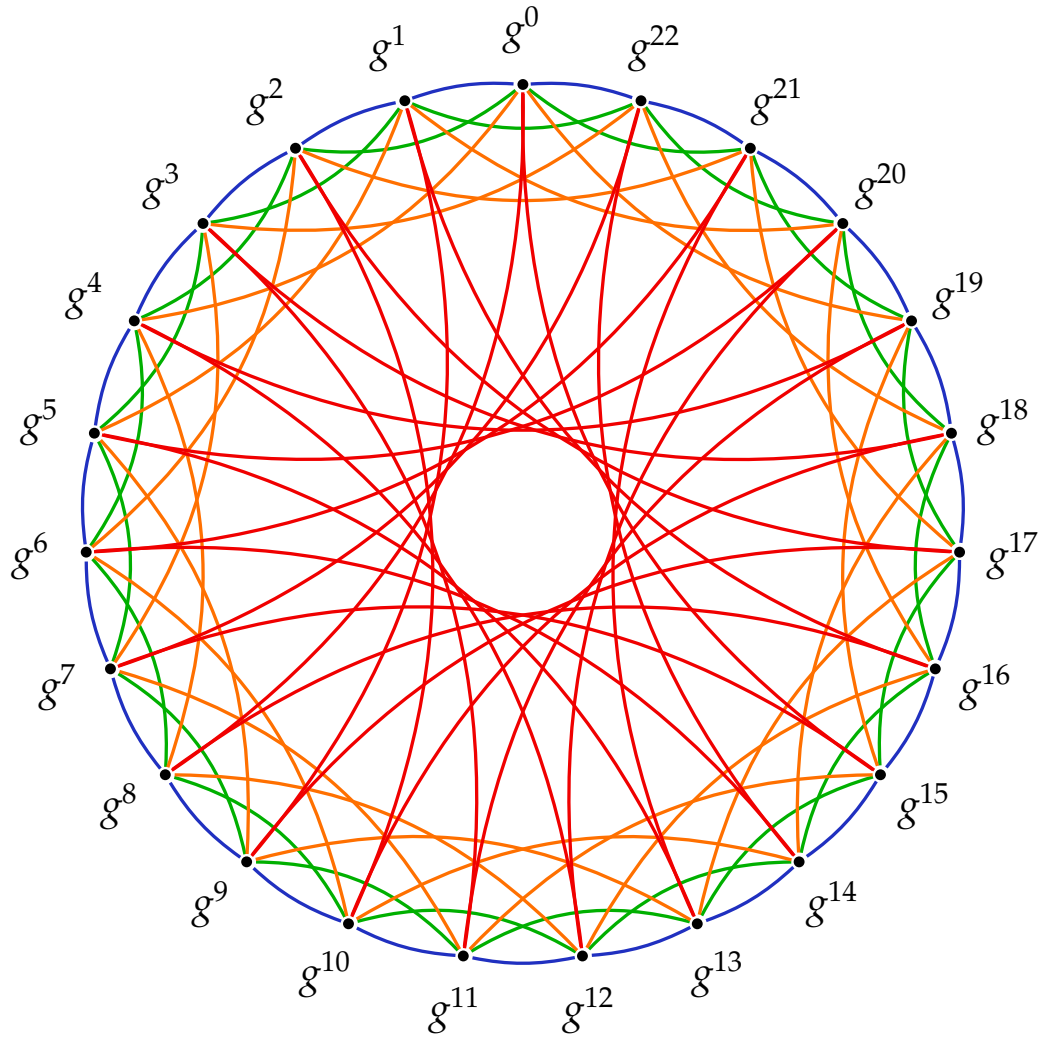
Square-and-multiply as graphs



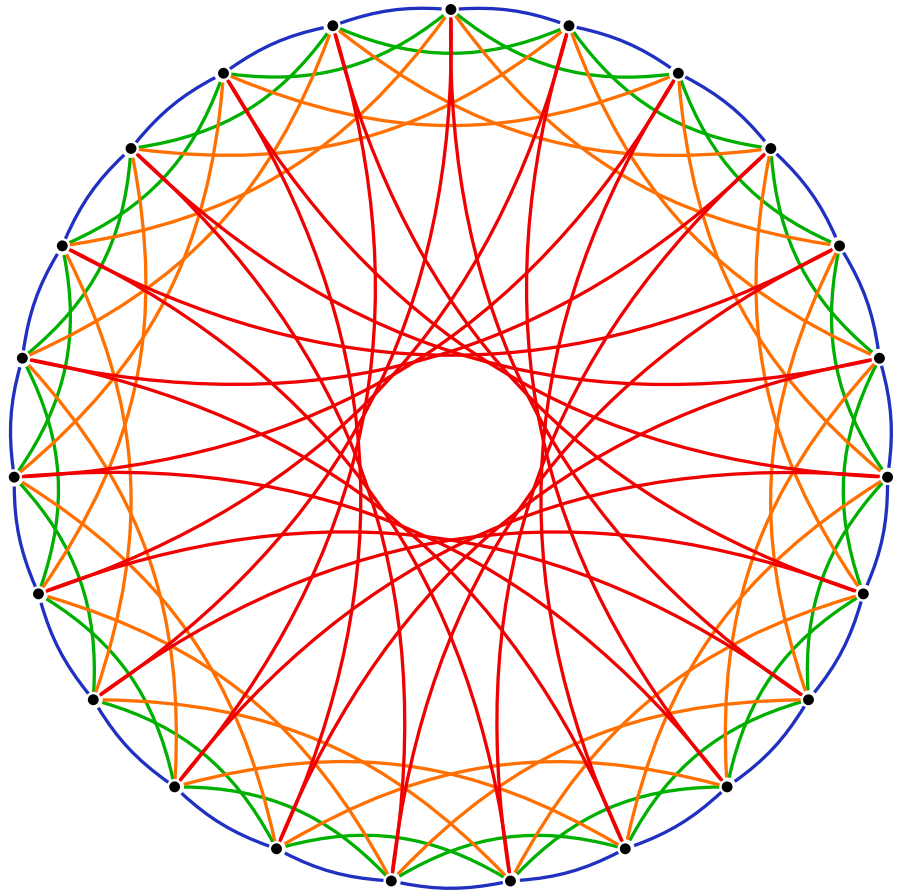
Square-and-multiply as graphs



Square-and-multiply as a graph



Square-and-multiply as a graph



Fast mixing: paths of length log(# nodes) to everywhere.

Shor's algorithm vs. DLP

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

Shor's algorithm vs. DLP

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

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

Shor's algorithm vs. DLP

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

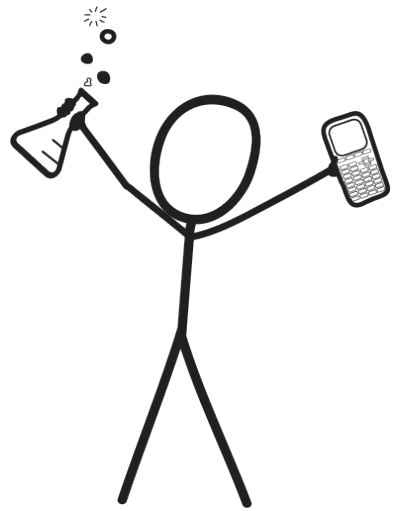
$$f\colon\ \mathbb{Z}^2 \to G,\ (x,y) \mapsto g^x \cdot h^y\ .$$

$\rightsquigarrow$ <u>New plan:</u> Get rid of the group, keep the graph.

Plan for this talk

- Some high-level intuition. ✓
- Elliptic curves & isogenies.
- Isogeny group actions.
- Signatures from isogenies.

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

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

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

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

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

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

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

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

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.

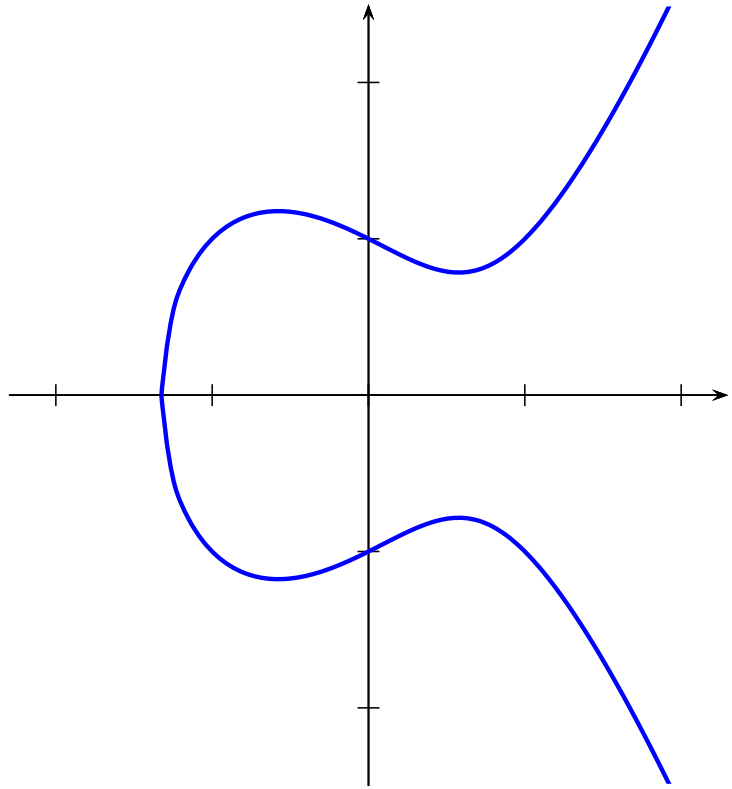
- The neutral element is ∞.
- The inverse of (x, y) is $(x, -y)$.
- The sum of (x_1, y_1) and (x_2, y_2) is

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

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.

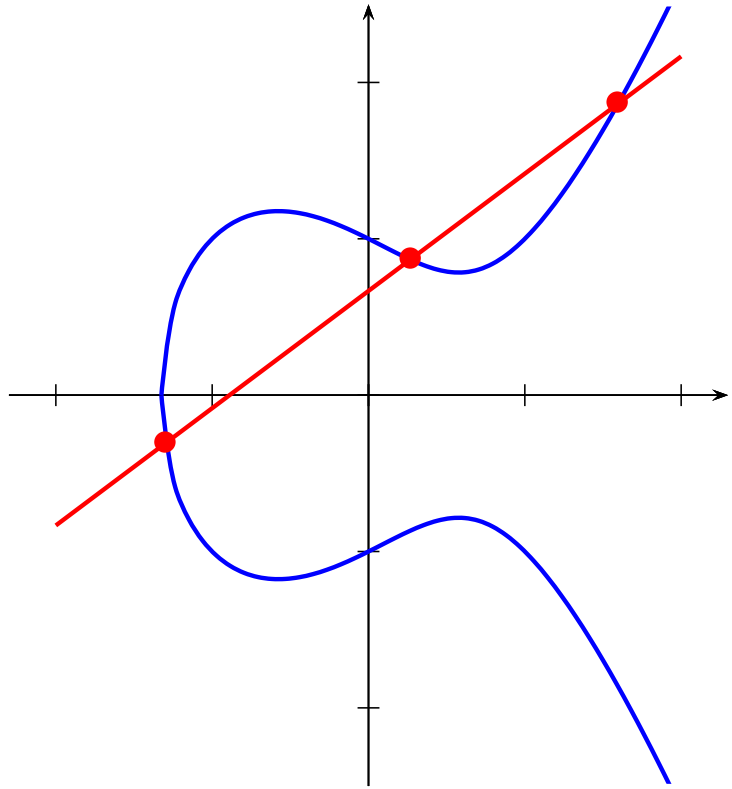
do **not** remember these formulas!

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



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

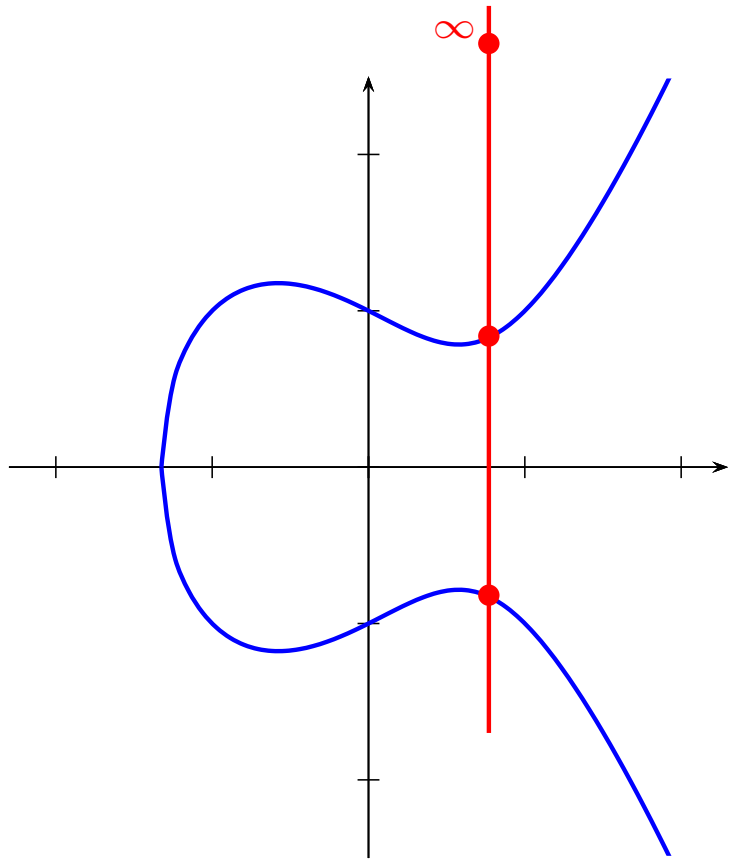
Elliptic curves (picture over $\mathbb{R}$)



Addition law:

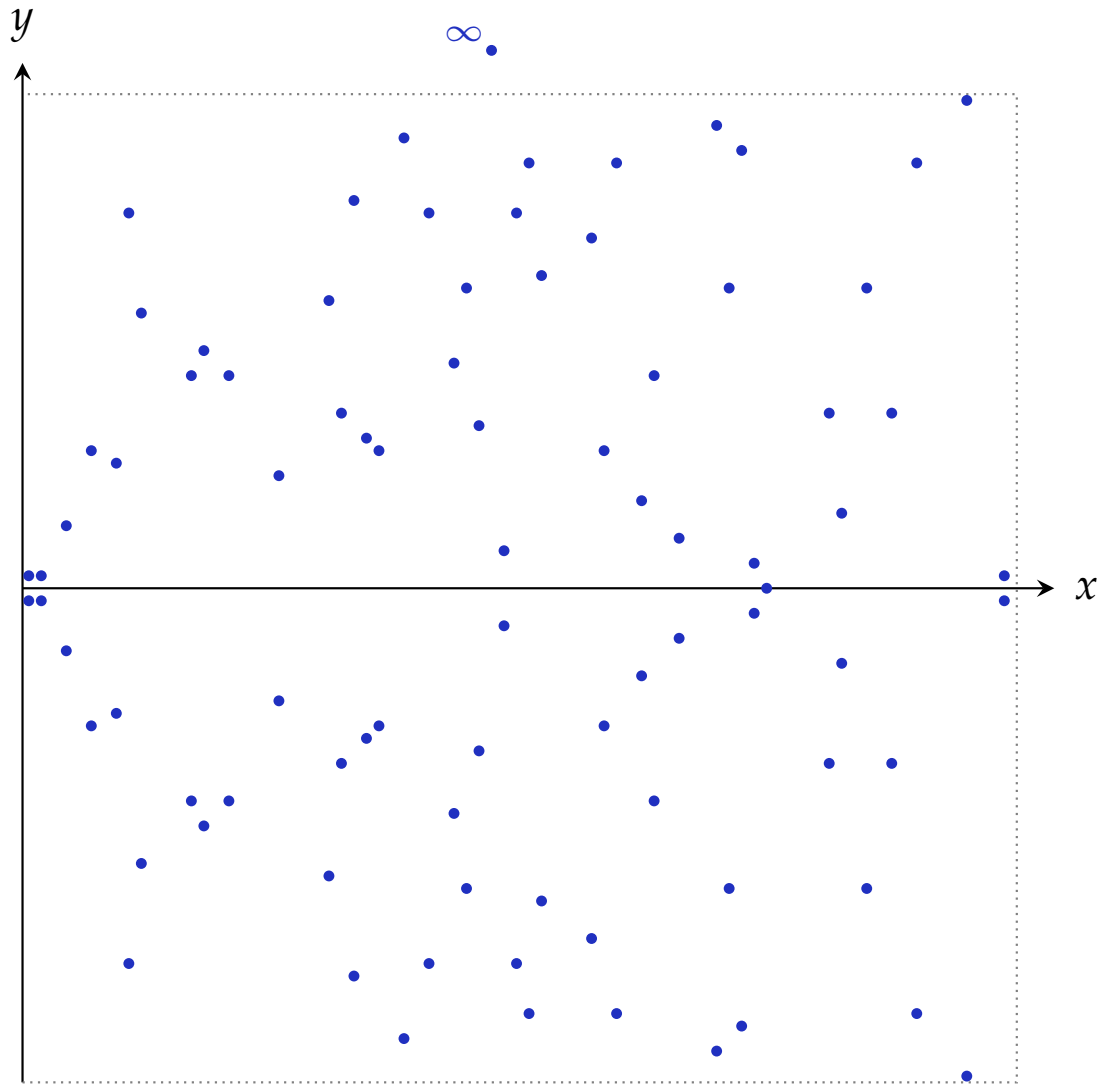
$$P + Q + R = \infty \iff \{P, Q, R\} \text{ on a straight line.}$$

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



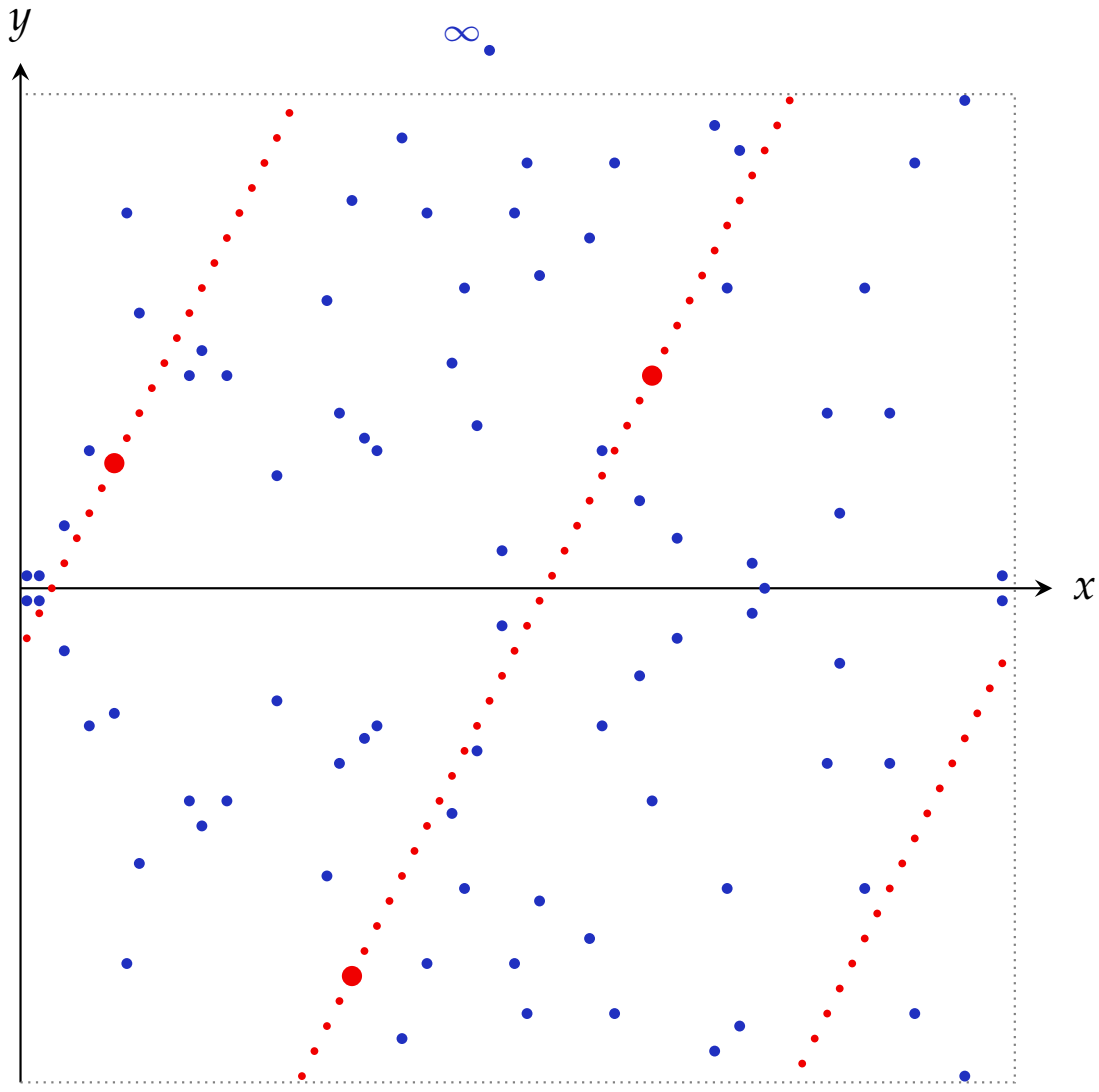
The *point at infinity* ∞ lies on every vertical line.

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 $\mathbb{F}_{79}$.

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)

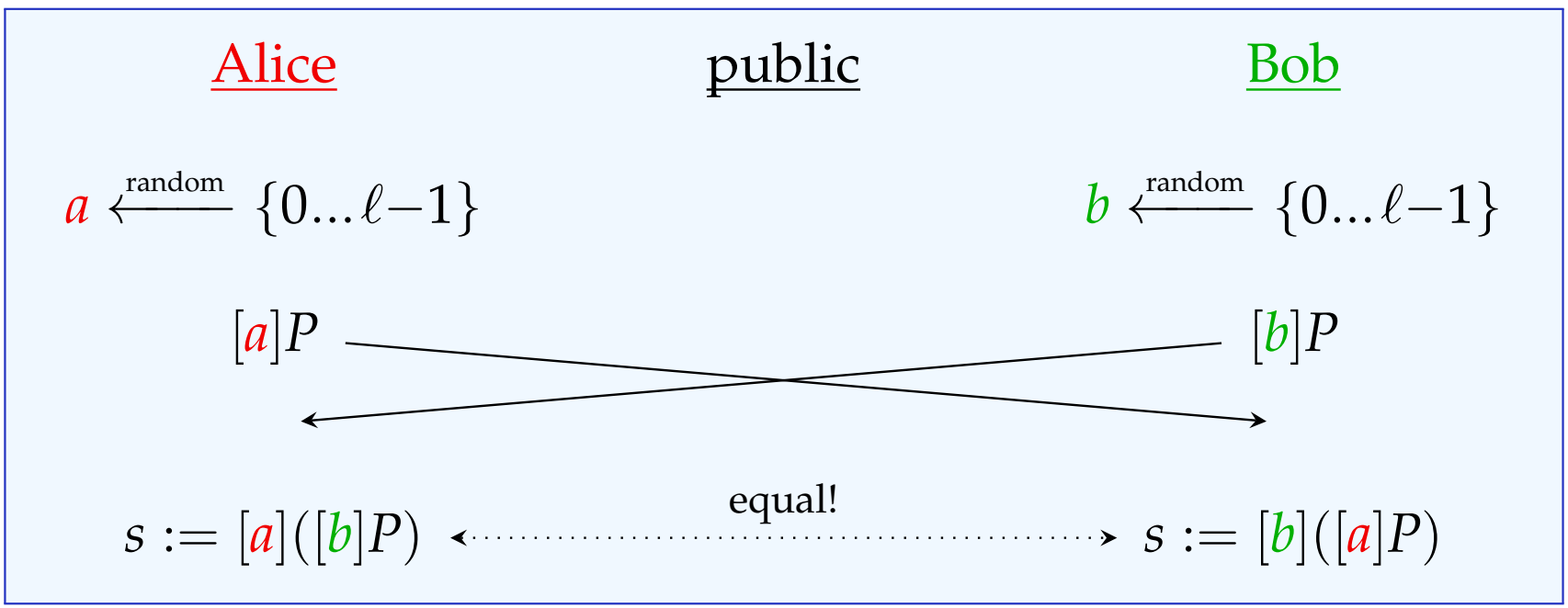
Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

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!)



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.

(This matches the degree of φ in x when written in lowest terms.)

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

$$[m] : E \to E$$

is a degree-m^2 isogeny. Notation: $E[m] := \ker[m]$.

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

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

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

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

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 $\pi \circ \varphi = \varphi \circ \pi$.

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 $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

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!)

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* separable* isogeny $\varphi_G : E \to E'$ with kernel G.

Isogenies and kernels

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

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

Isogenies and kernels

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

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

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

Isogenies and kernels

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

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

$\rightsquigarrow$ 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!)

Isogenies and kernels

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

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

$\rightsquigarrow$ 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.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \left(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q)\big), \right.$$
$$\left. y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q)\big) \right)$$

defines an isogeny of elliptic curves with kernel G.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \left(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q)\big), \right.$$
$$\left. y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q)\big) \right)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \left(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q)\big), \right.$$
$$\left. y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q)\big) \right)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

Complexity: $\Theta(\#G) \rightsquigarrow$ only suitable for small degrees.

The $\sqrt{\text{élu}}$ algorithm reduces the cost to $\widetilde{\mathcal{O}}(\sqrt{\#G})$.

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, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ or $E(\mathbb{F}_p) \cong \mathbb{Z}/\frac{p+1)}{2} \times \mathbb{Z}/2$, 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, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ or $E(\mathbb{F}_p) \cong \mathbb{Z}/\frac{p+1)}{2} \times \mathbb{Z}/2$, 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, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ or $E(\mathbb{F}_p) \cong \mathbb{Z}/\frac{p+1)}{2} \times \mathbb{Z}/2$, 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 for the rest of the lecture.)

“Computing an isogeny”

“Computing an isogeny”

Keep in mind: Constructing isogenies $E \to \_$ is (usually) easy, constructing an isogeny $E \to E'$ given (E, E') is (usually) hard.

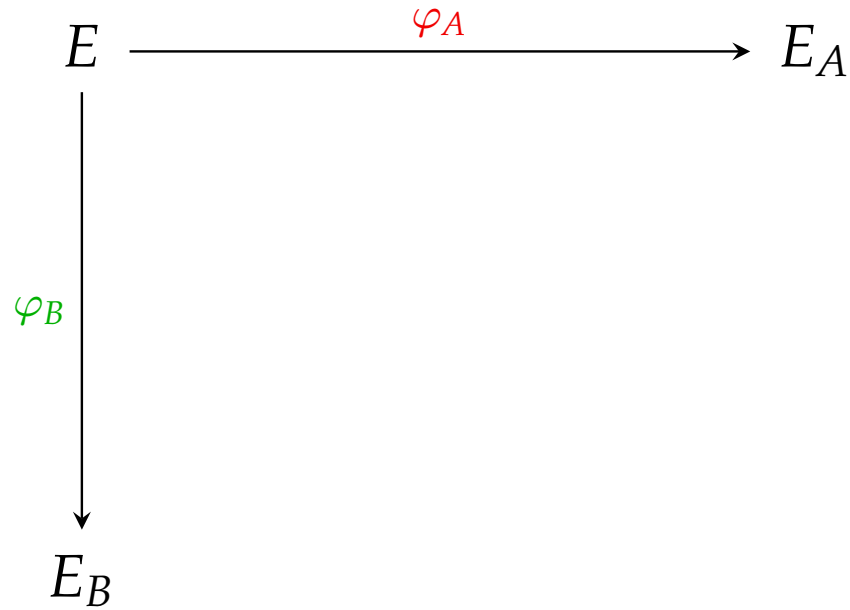
Plan for this talk

- Some high-level intuition. ✓
- Elliptic curves & isogenies. ✓
- Isogeny group actions.
- Signatures from isogenies.

Isogeny-based key exchange: High-level view

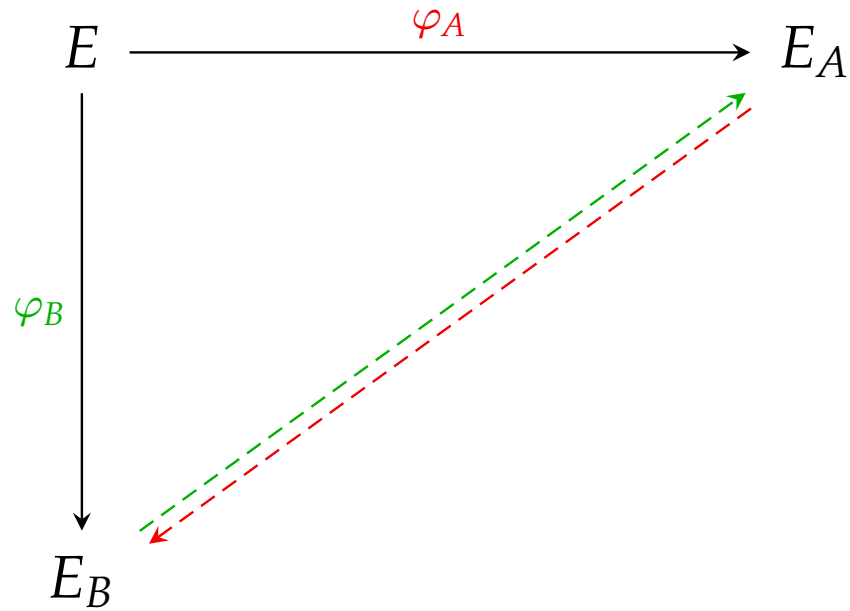
E

Isogeny-based key exchange: High-level view



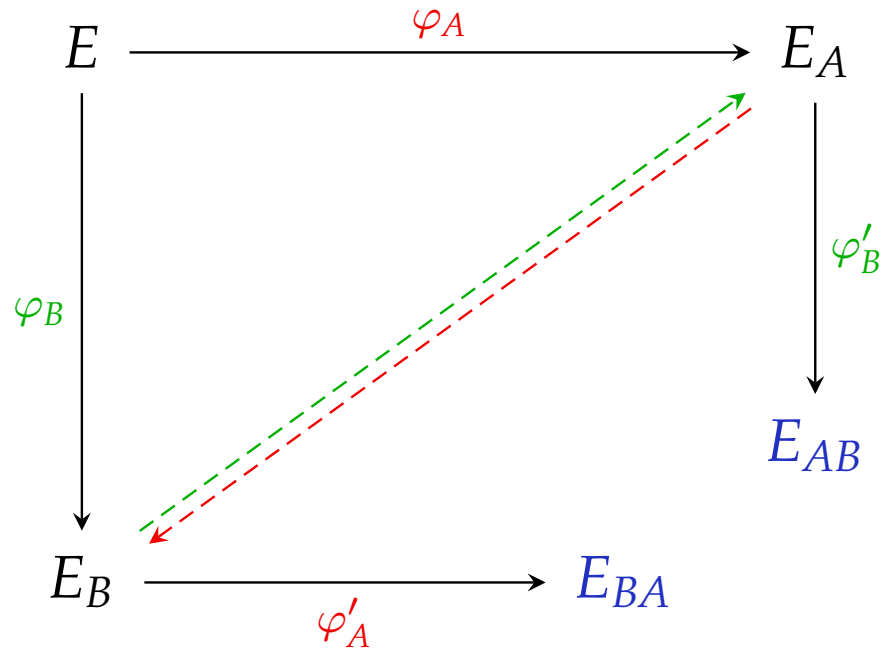
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)

Isogeny-based key exchange: High-level view



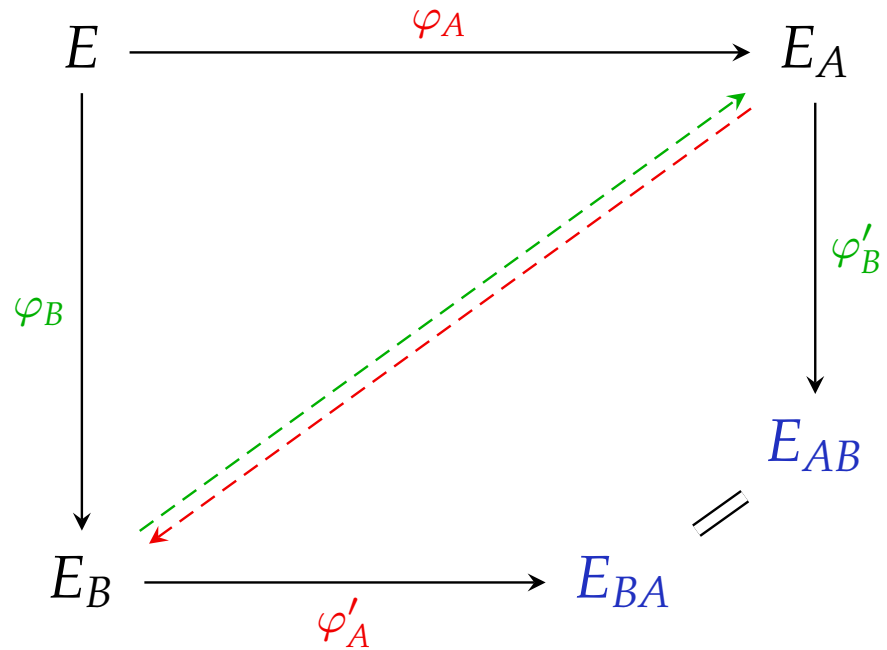
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.

Isogeny-based key exchange: High-level view



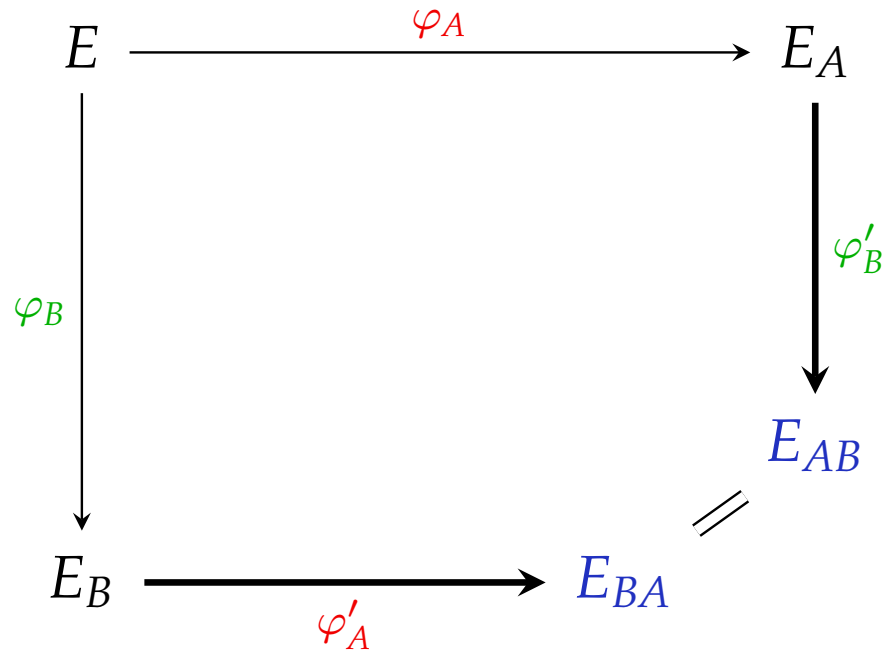
- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice somehow finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob somehow finds $\varphi_{B'} :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

Isogeny-based key exchange: High-level view

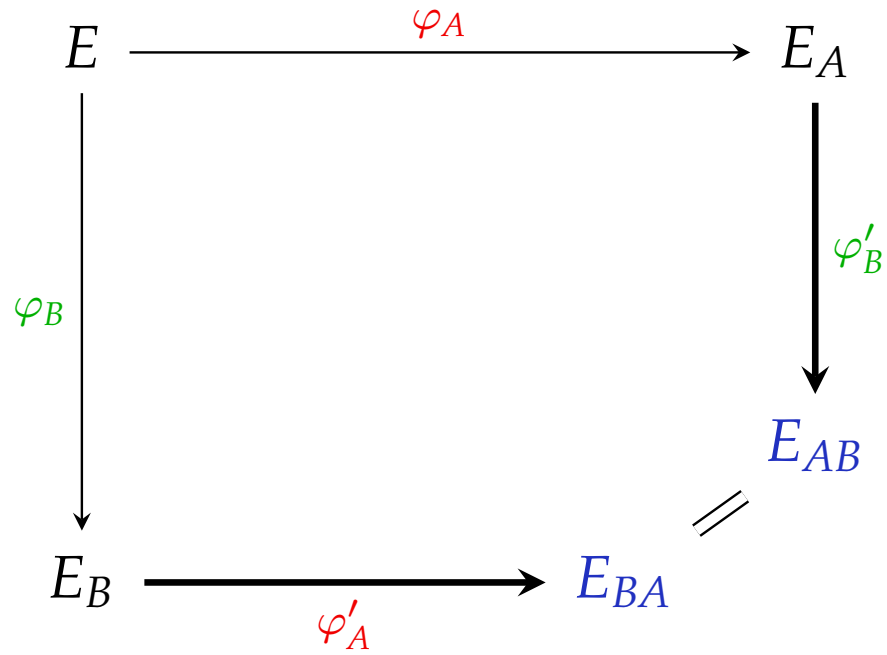


- Alice & Bob pick secret $\varphi_A : E \to E_A$ and $\varphi_B : E \to E_B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the end curves E_A and E_B.
- Alice somehow finds a "parallel" $\varphi_{A'} : E_B \to E_{BA}$, and Bob somehow finds $\varphi_{B'} :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

How to find “parallel” isogenies?



How to find "parallel" isogenies?



CSIDH's solution:

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.)


CSIDH ['si:ˌsaɪd]
[Castryck–Lange–Martindale–Panny–Renes 2018]

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

"Special" isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

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

“Special” isogenies

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

$\Rightarrow$ 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

We fix an elliptic curve $E/\mathbb{F}_p$ such that $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$.

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

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

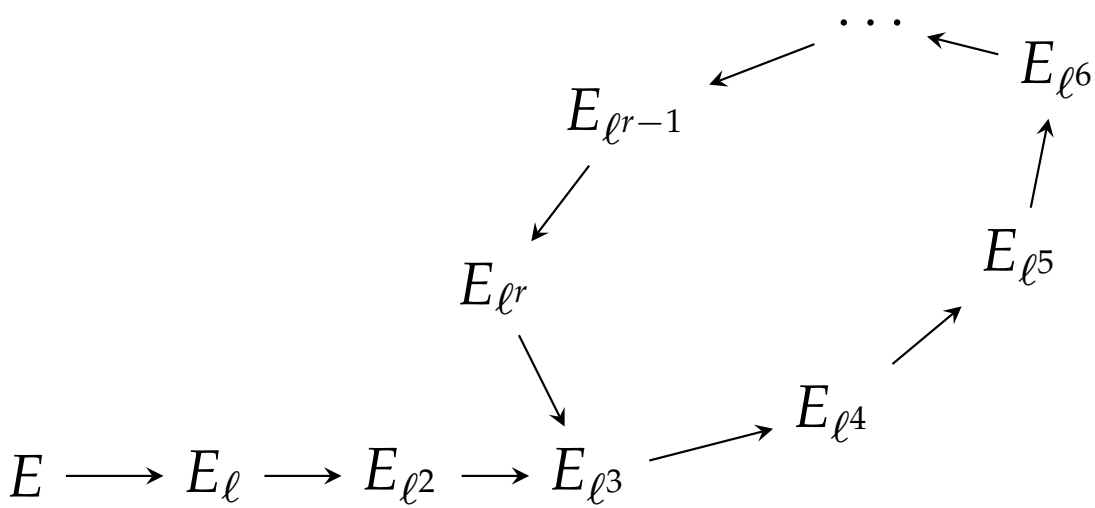
We consider prime ℓ and refer to φ_ℓ as a "special" isogeny.

Cycles from “special” isogenies

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

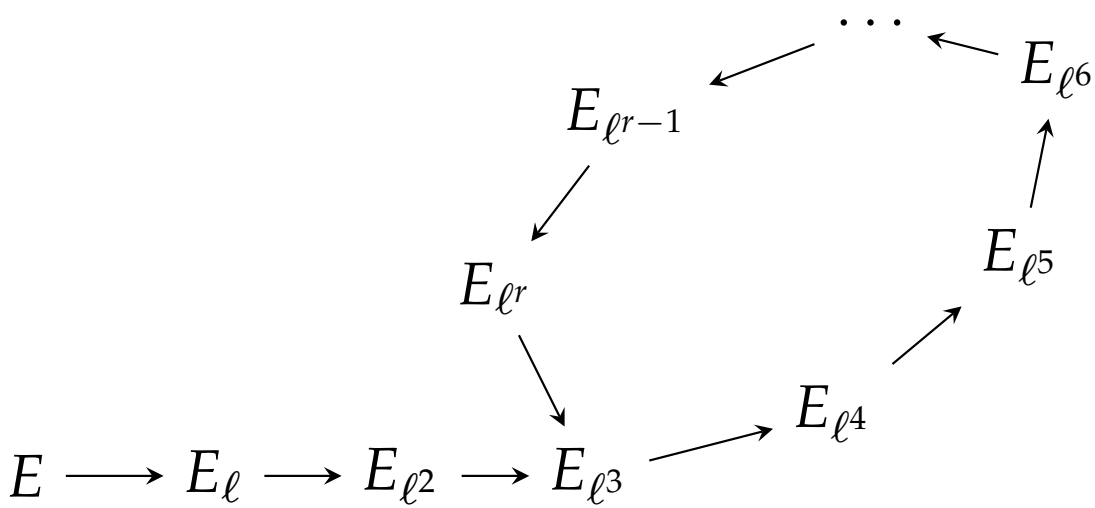
Cycles from "special" isogenies

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



Cycles from "special" isogenies

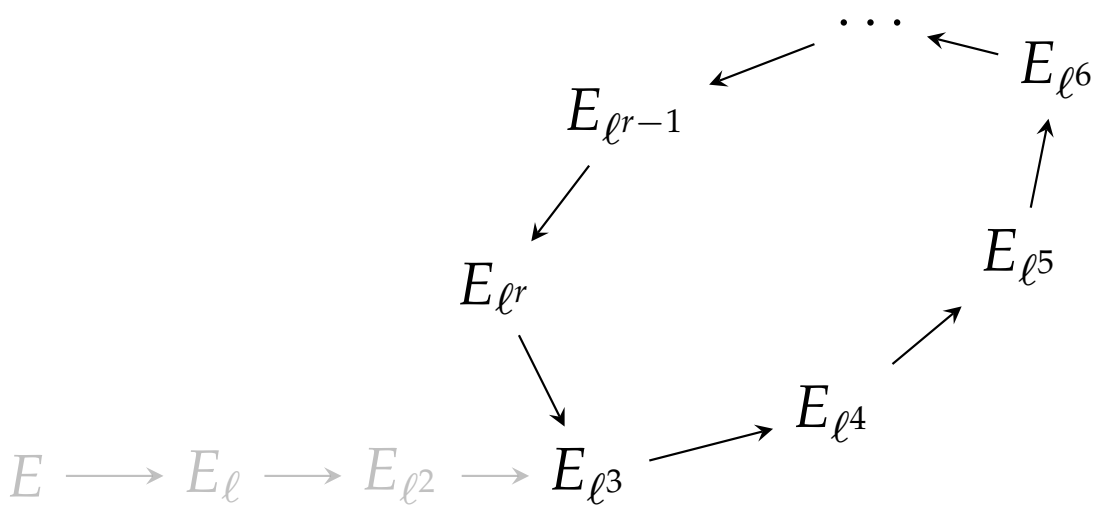
What happens when we iterate such a "special" isogeny?



- Fact: Each curve has only one other rational ℓ-isogeny.
  It is defined by the kernel $\{P \in E(\mathbb{F}_{p^2}) : [\ell_i]P = \infty \wedge \pi(P) = -P\}$.

Cycles from "special" isogenies

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

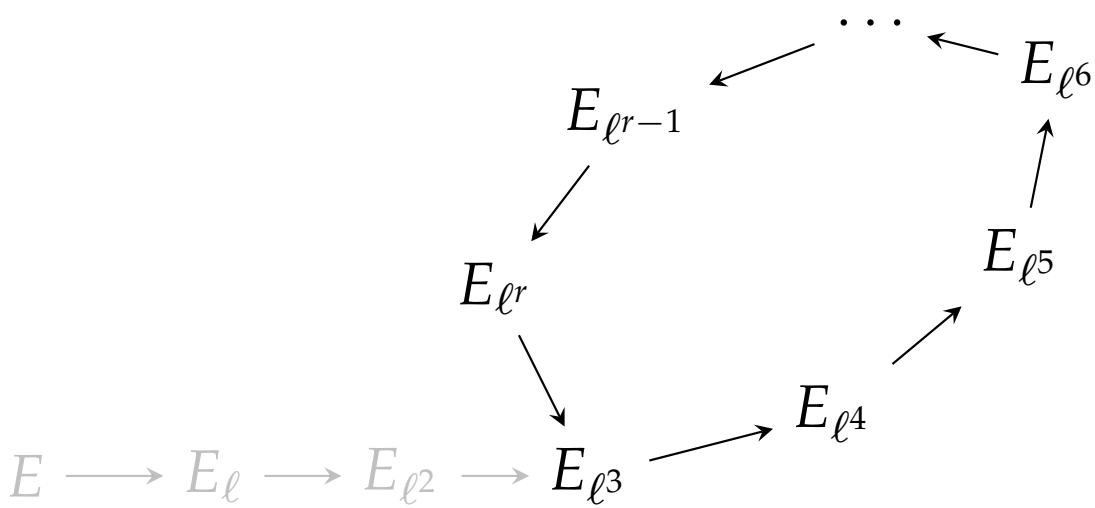


- Fact: Each curve has only one other rational ℓ-isogeny.
  It is defined by the kernel $\{P \in E(\mathbb{F}_{p^2}) : [\ell_i]P = \infty \wedge \pi(P) = -P\}$.

!! 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?



- Fact: Each curve has only one other rational ℓ-isogeny.
  It is defined by the kernel $\{P \in E(\mathbb{F}_{p^2}) : [\ell_i]P = \infty \wedge \pi(P) = -P\}$.

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

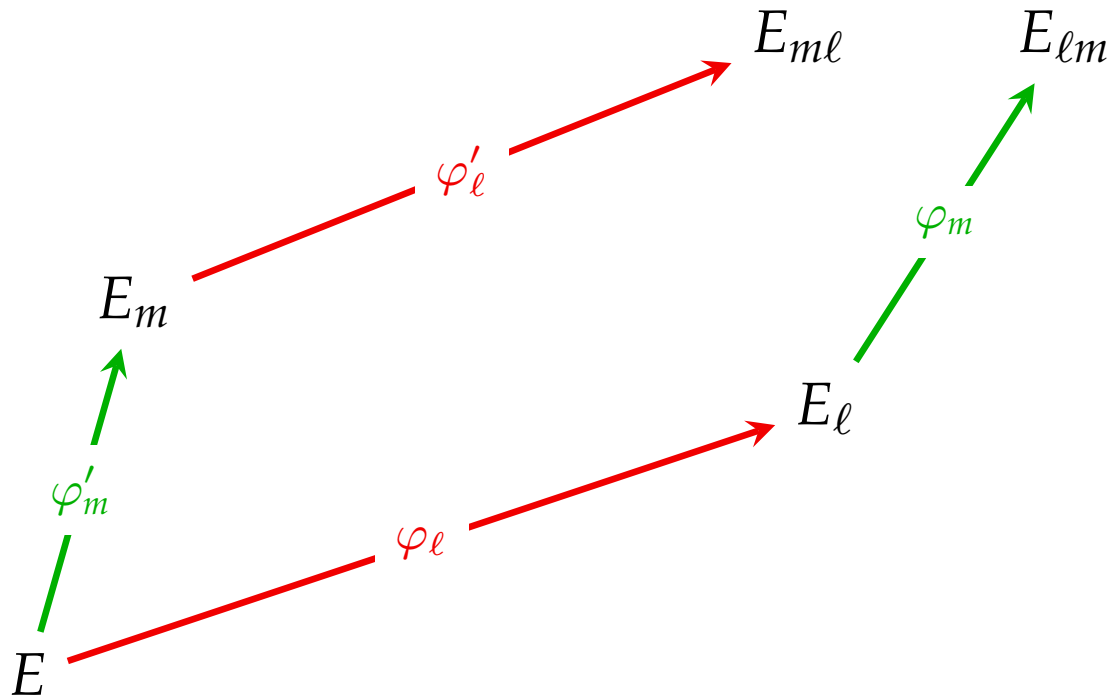
$\Longrightarrow$ The "special" isogenies φ_ℓ form isogeny cycles!

Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?

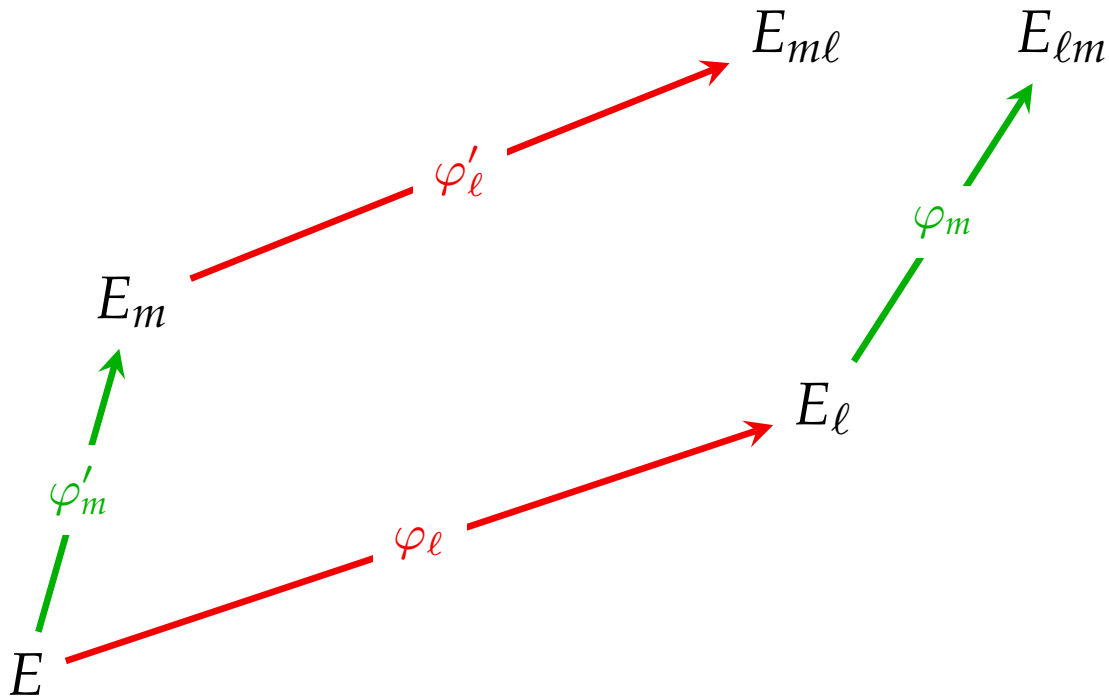
Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?



Compatible cycles from "special" isogenies

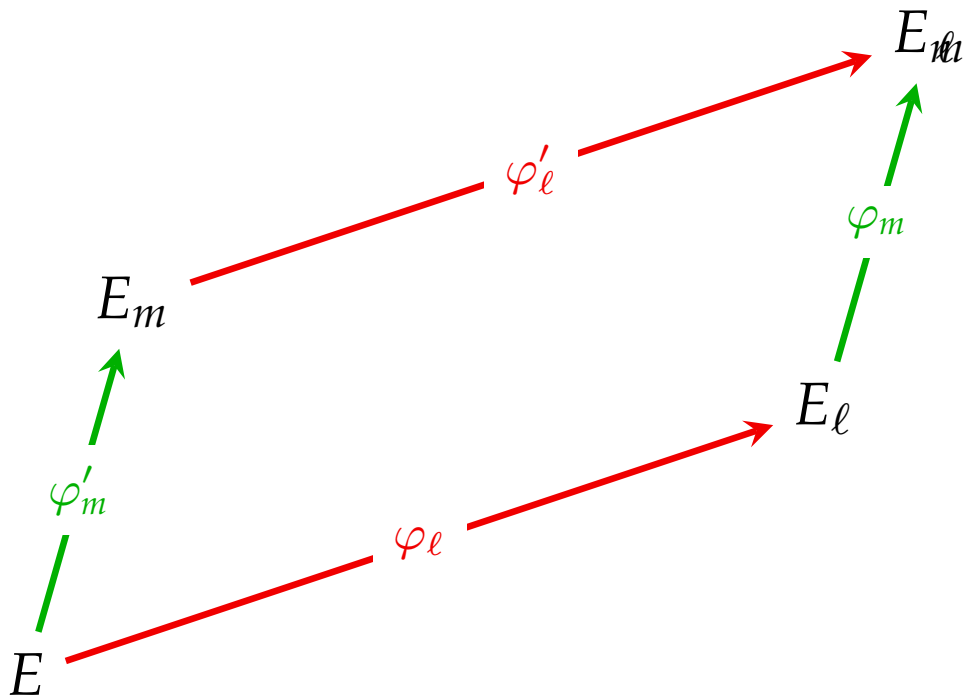
What happens when we compose those "special" isogenies?



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

Compatible cycles from "special" isogenies

What happens when we compose those "special" isogenies?



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

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

CSIDH in one slide

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.

CSIDH in one slide

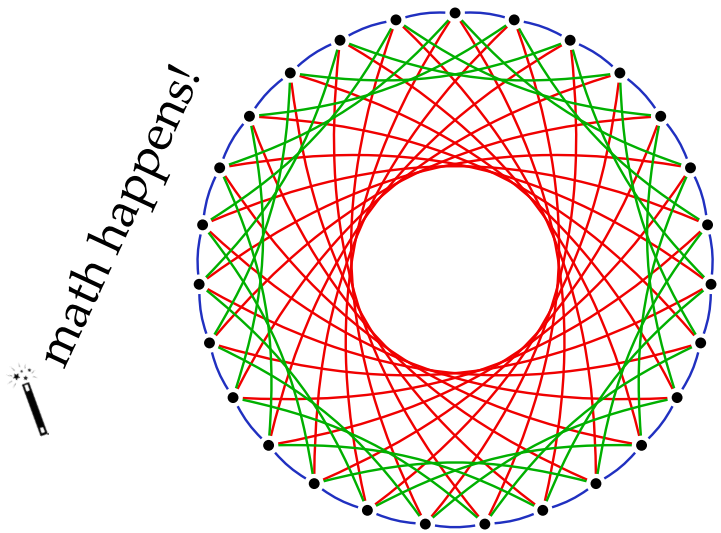
- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.

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

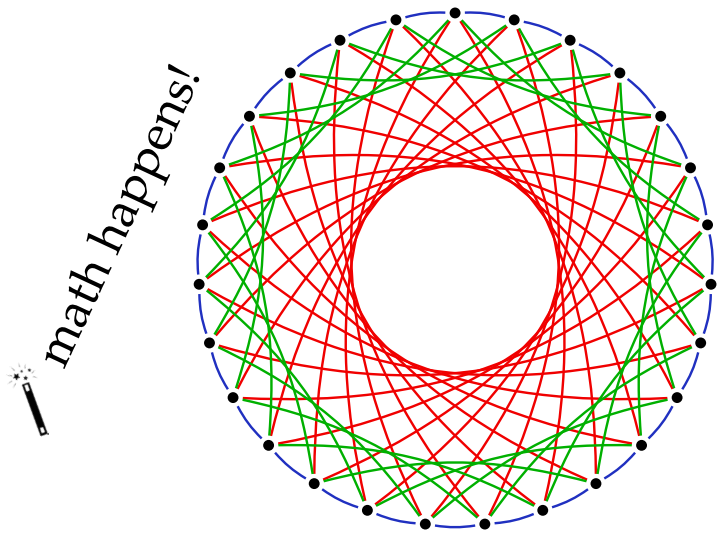
- Choose some small odd primes $\ell_1, \ldots, \ell_n$.
- Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
- Let $X = \{y^2 = x^3 + Ax^2 + x$ supersingular with $A \in \mathbb{F}_p\}$.
- Look at the "special" ℓ_i-isogenies within X.



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

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.

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.)

Walking in the CSIDH graph (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
```

Walking in the CSIDH graph (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)
```

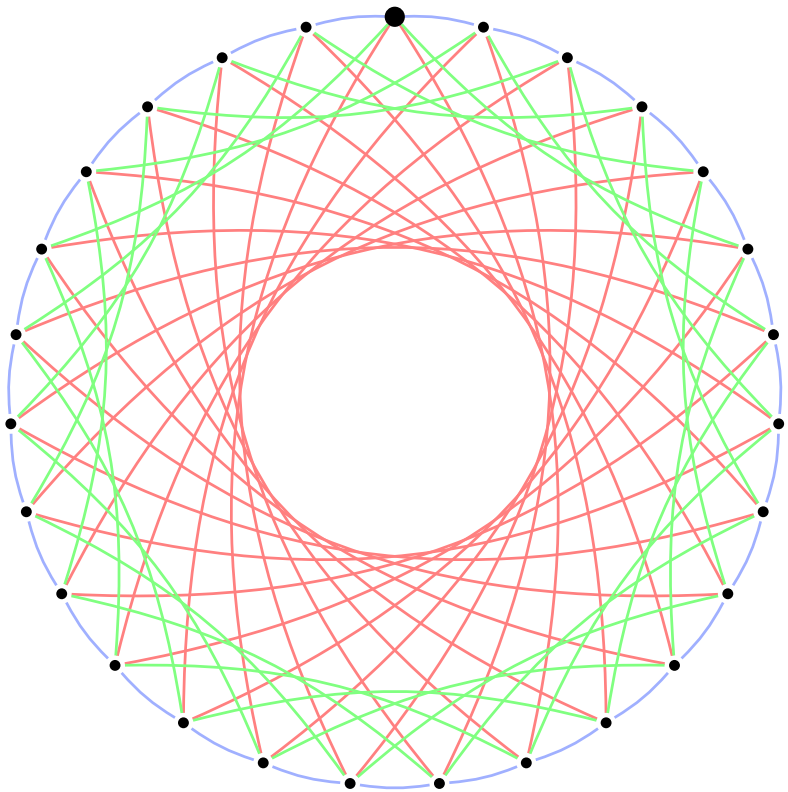
Walking in the CSIDH graph (in SageMath)

```
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
```

CSIDH key exchange

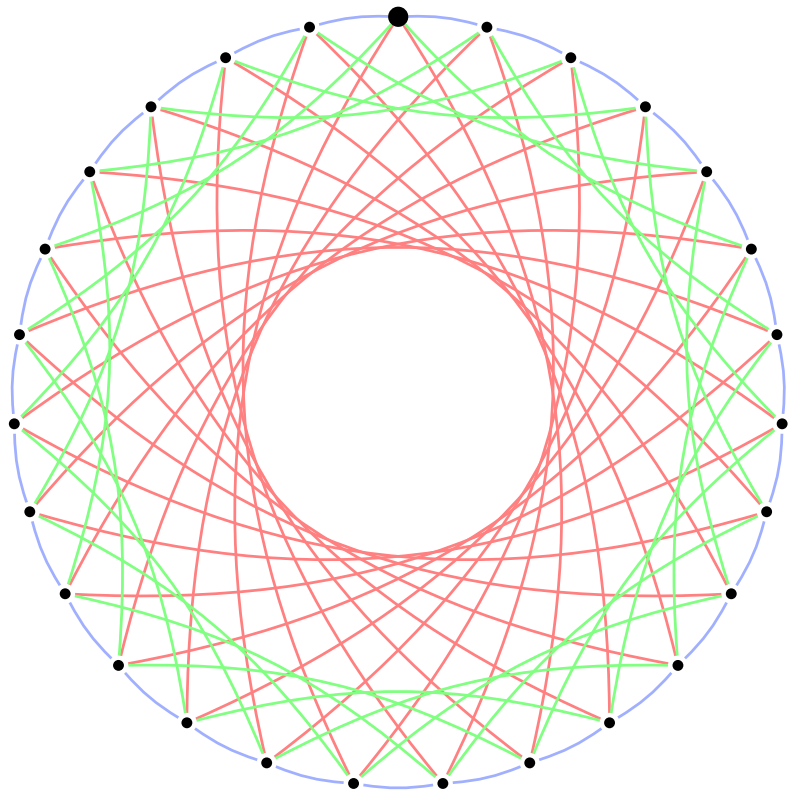
Alice

$[+, +, -, -]$



Bob

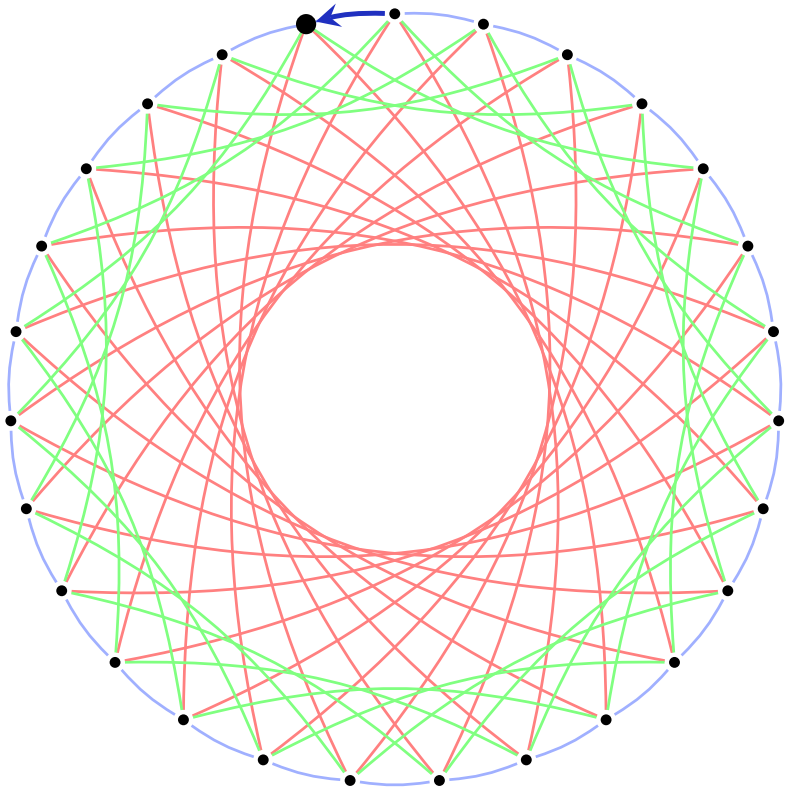
$[-, +, -, -]$



CSIDH key exchange

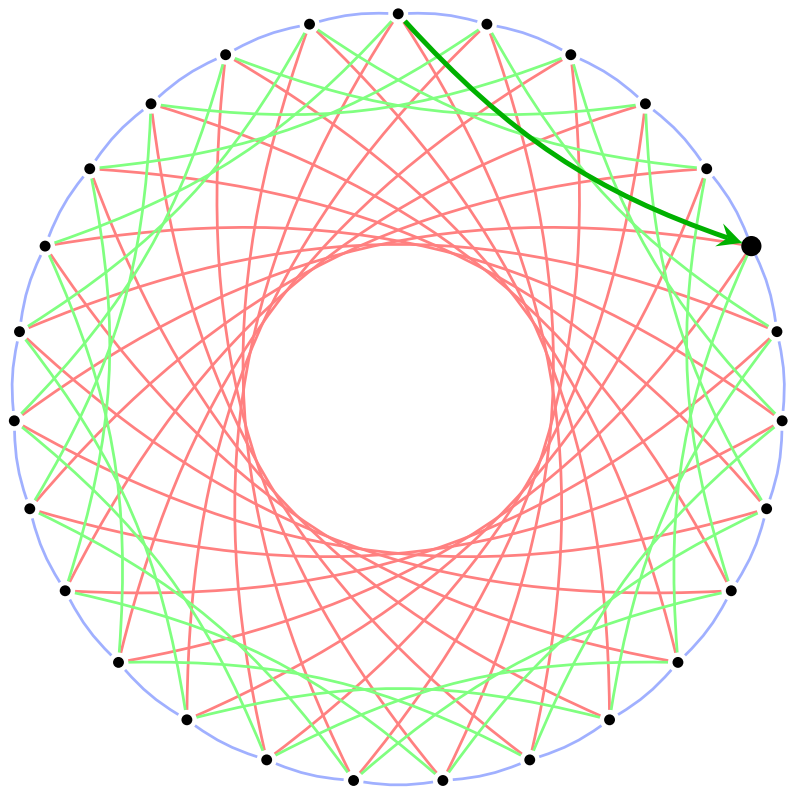
Alice

$[+, +, -, -]$



Bob

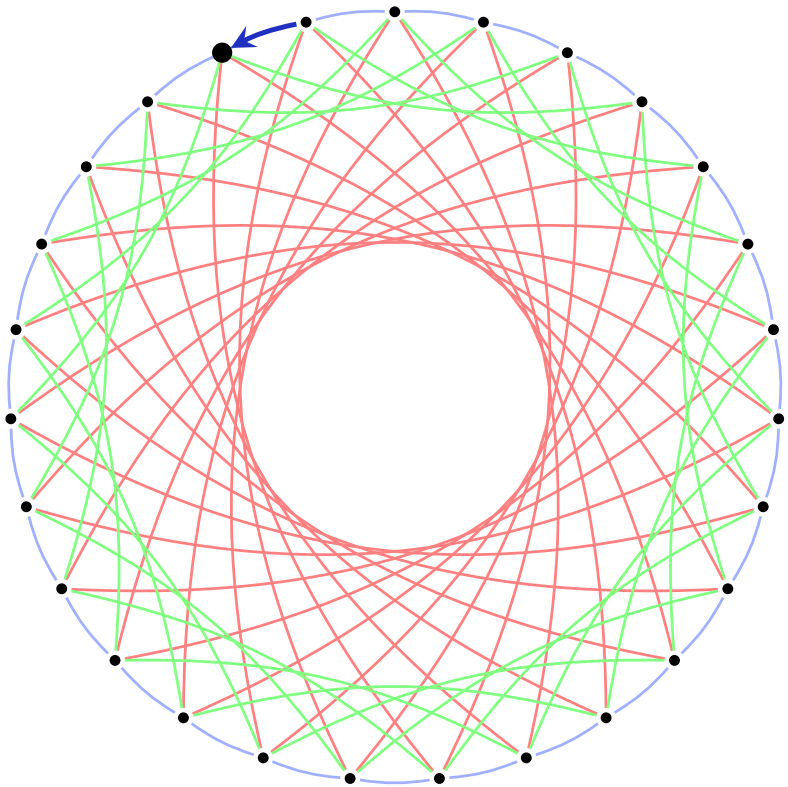
$[-, +, -, -]$



CSIDH key exchange

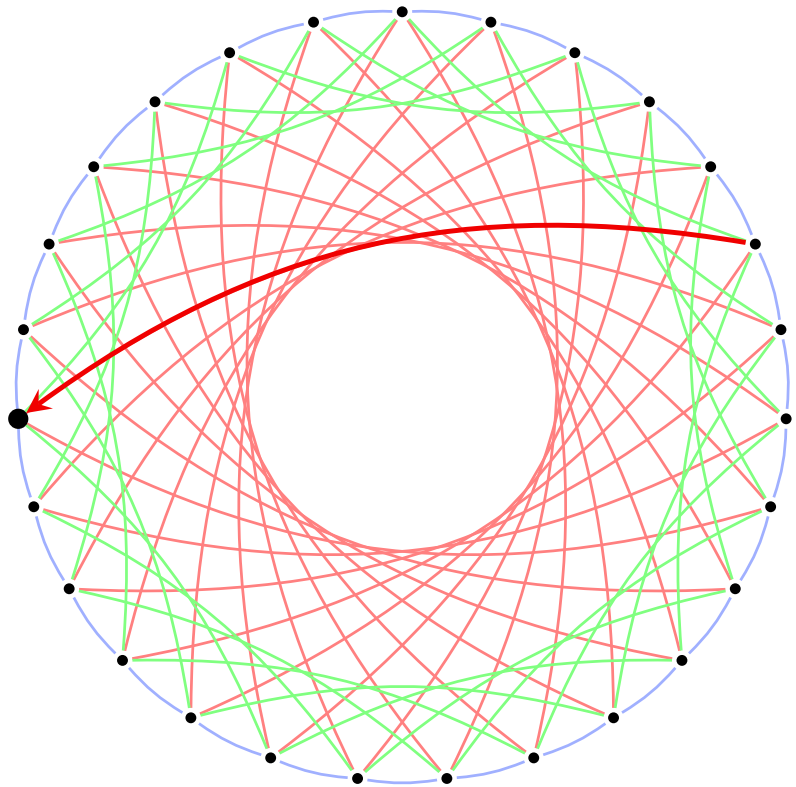
Alice

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



Bob

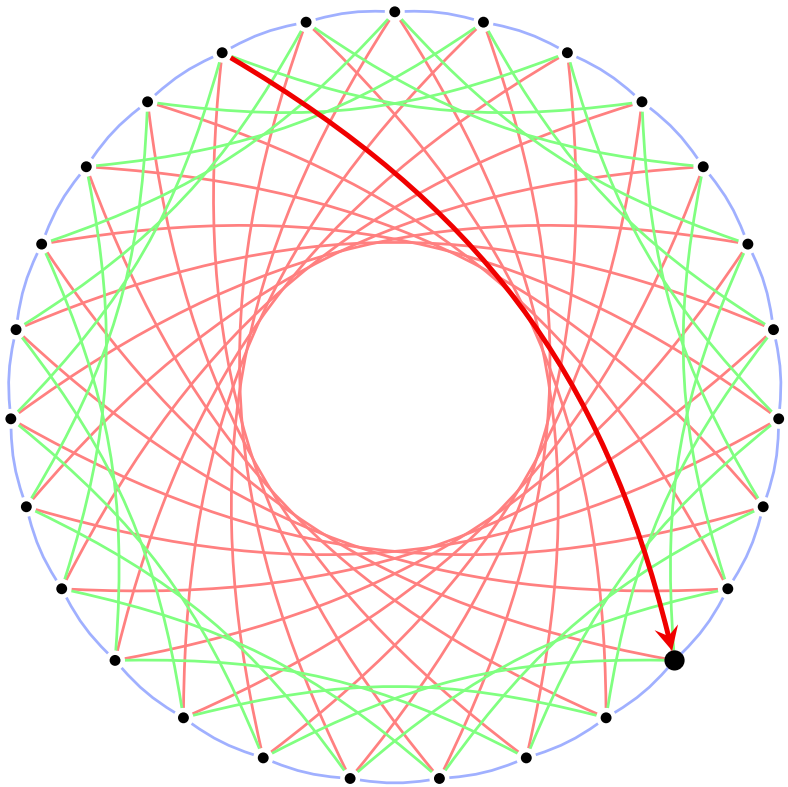
$[-, \underset{\uparrow}{+}, -, -]$



CSIDH key exchange

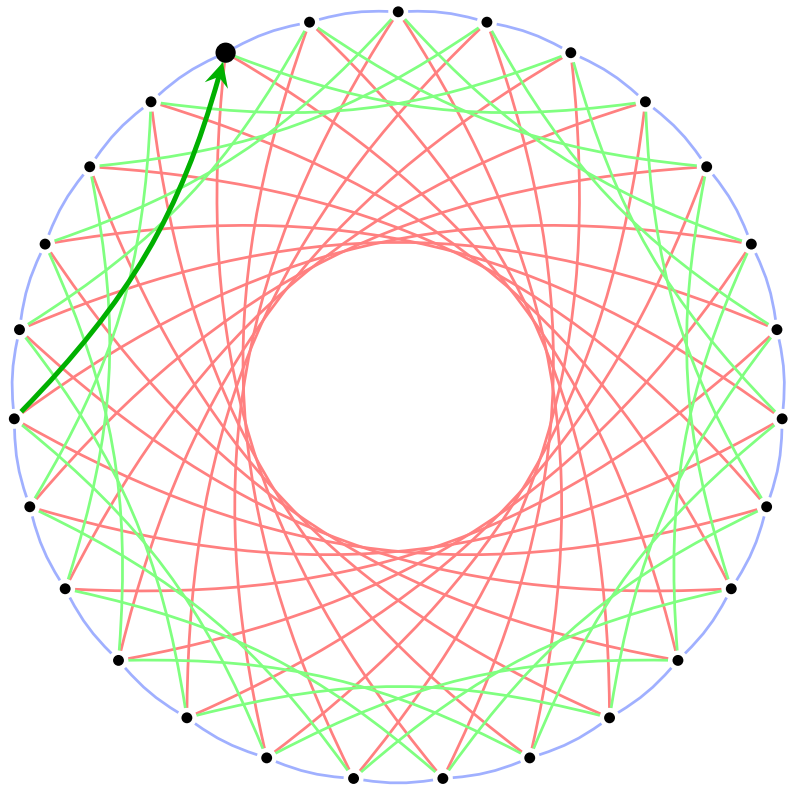
Alice

$[+, +, -, -]$



Bob

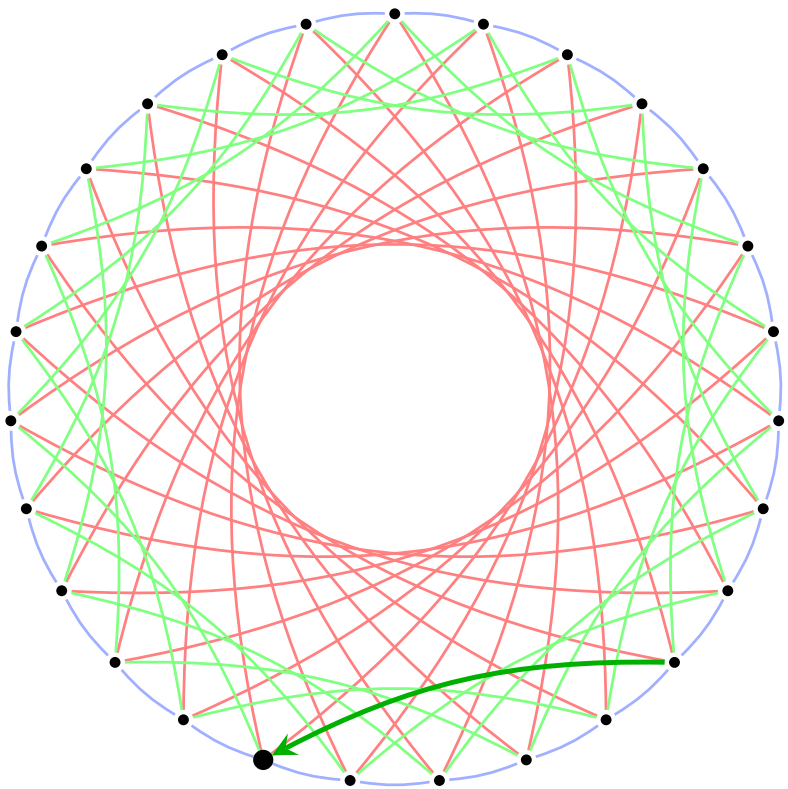
$[-, +, -, -]$



CSIDH key exchange

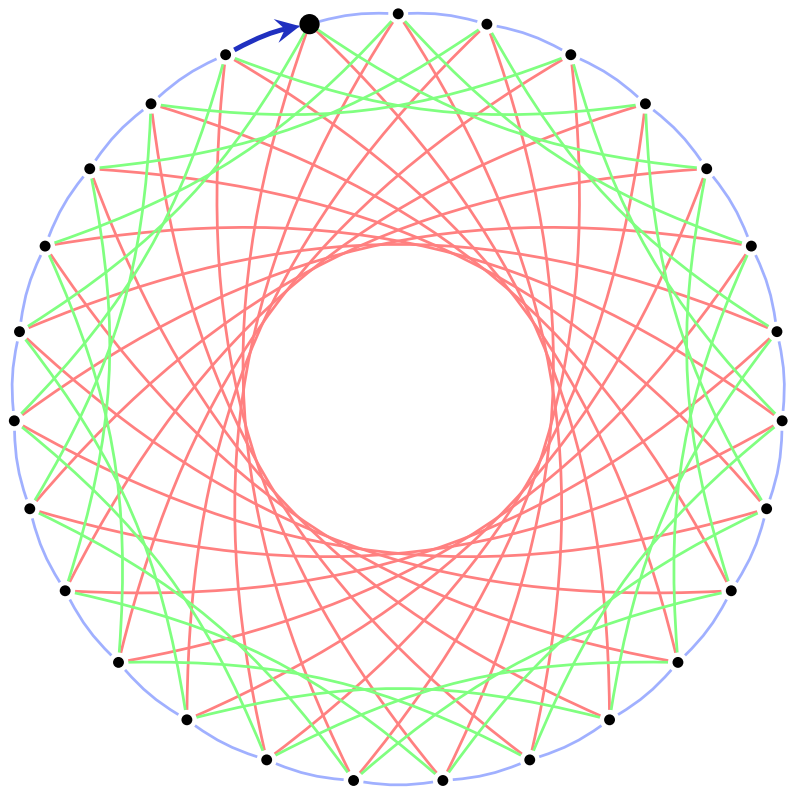
Alice

$[+, +, -, -]$



Bob

$[-, +, -, -]$



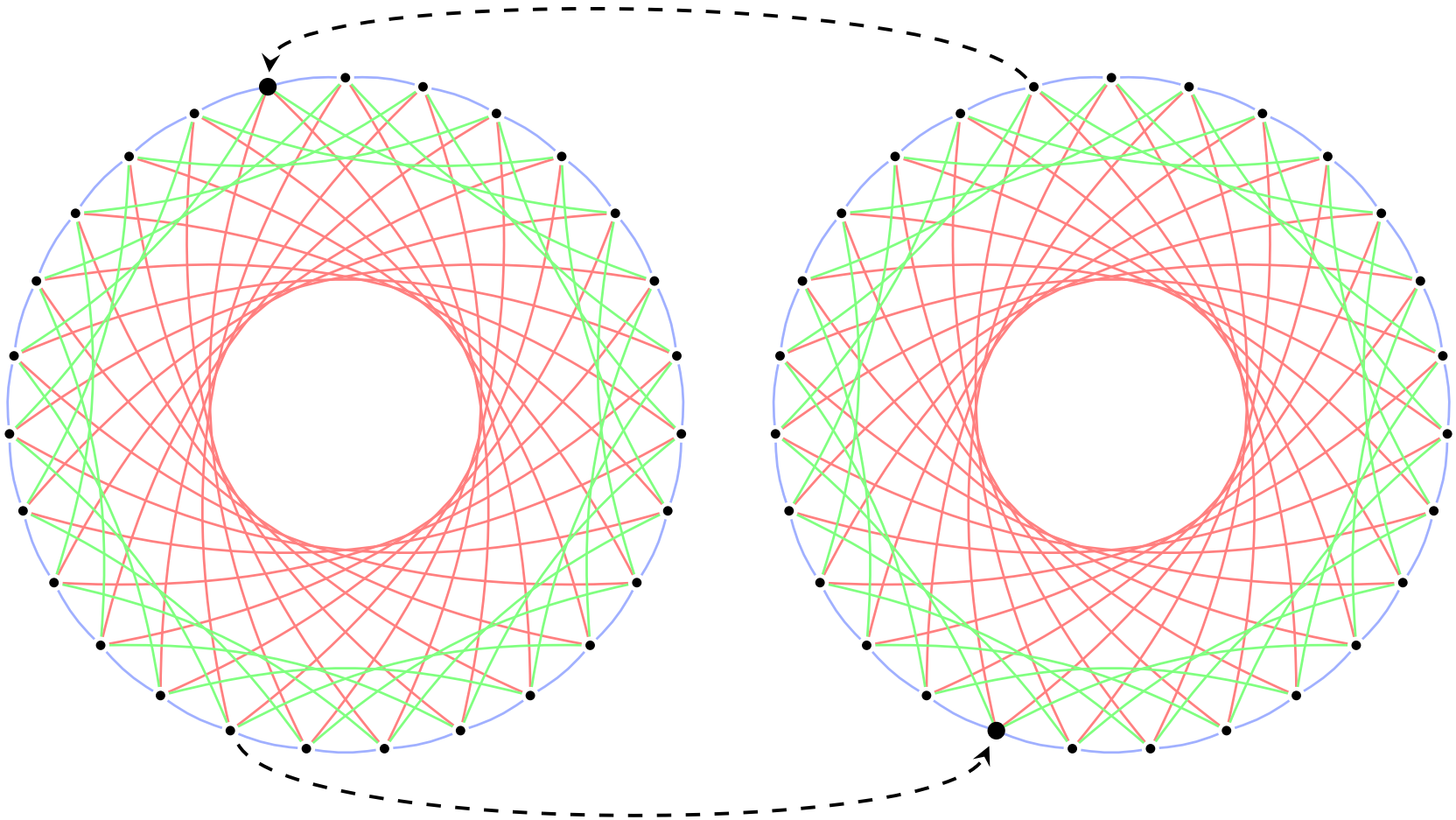
CSIDH key exchange

Alice

$[+, +, -, -]$

Bob

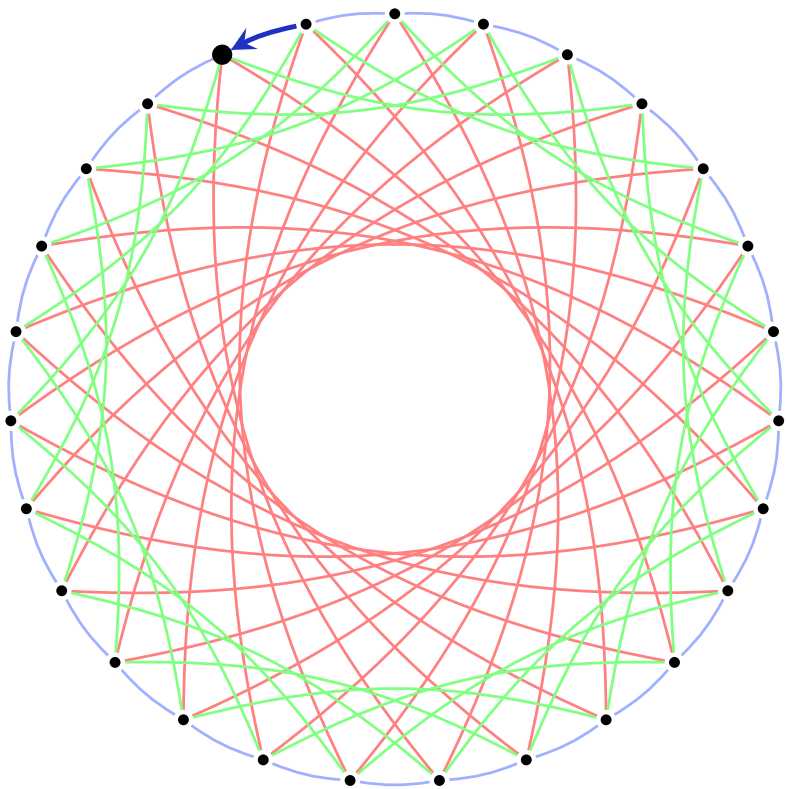
$[-, +, -, -]$



CSIDH key exchange

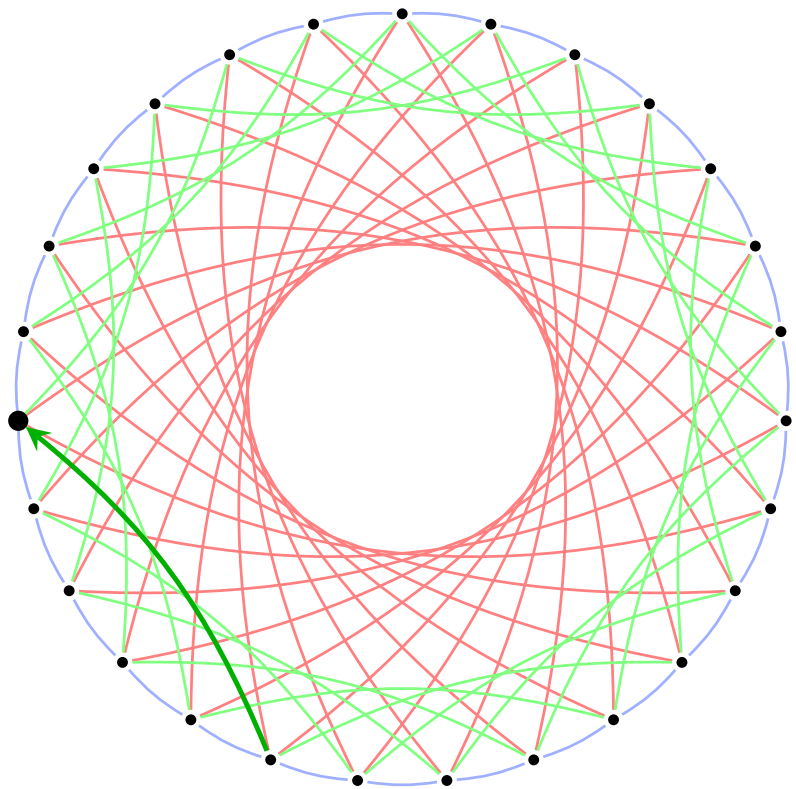
Alice

$[+, +, -, -]$



Bob

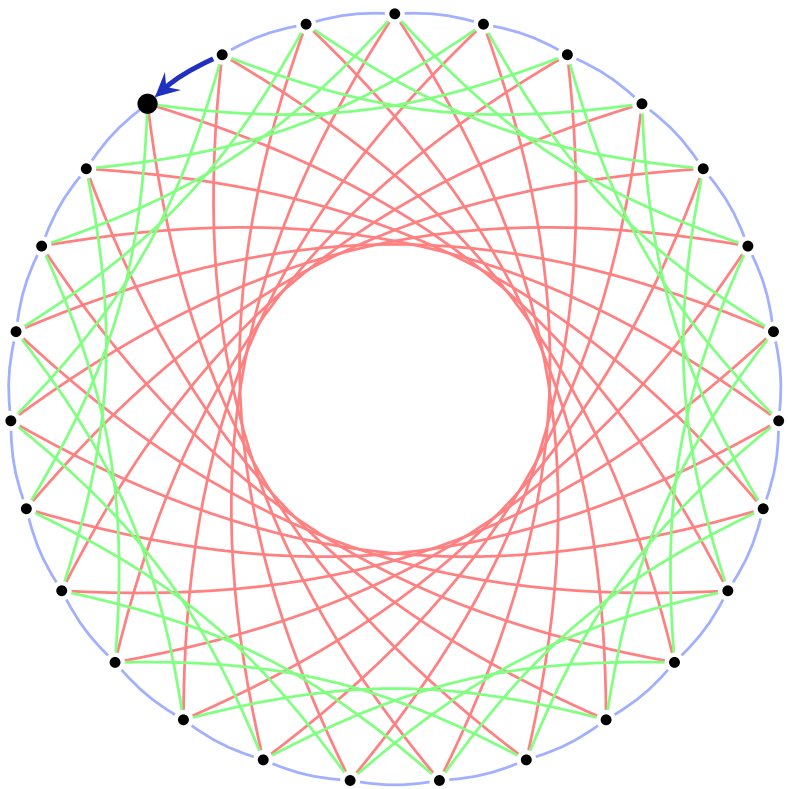
$[-, +, -, -]$



CSIDH key exchange

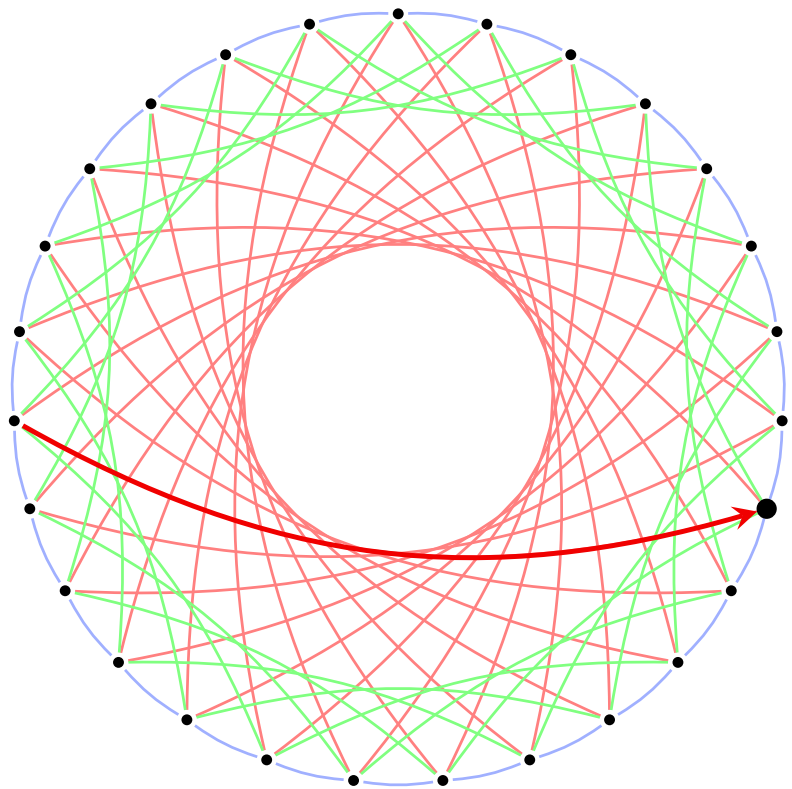
Alice

$[+, +, -, -]$



Bob

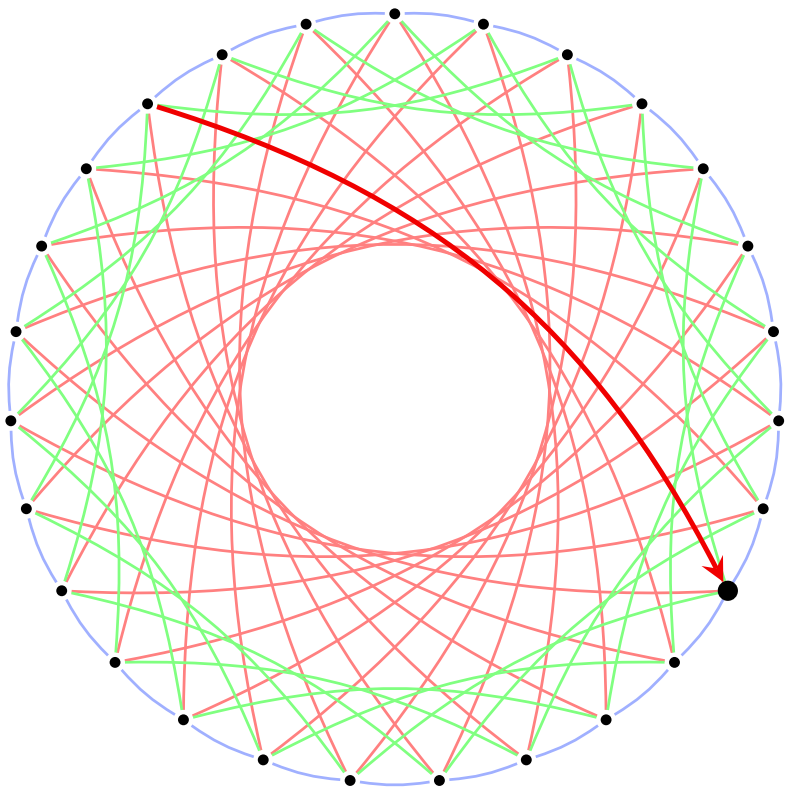
$[-, +, -, -]$



CSIDH key exchange

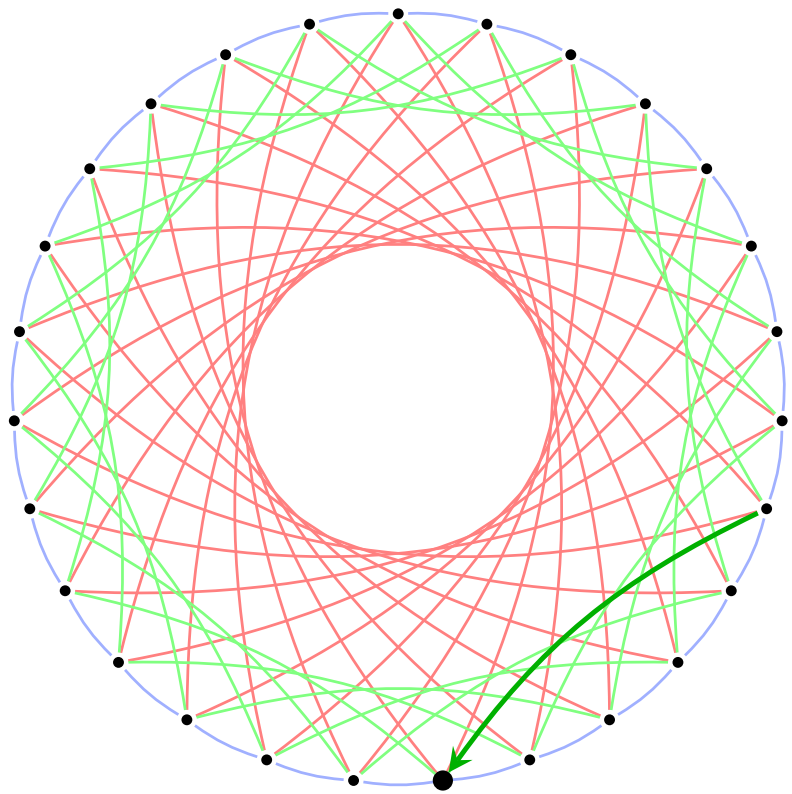
Alice

$[+, +, -, -]$



Bob

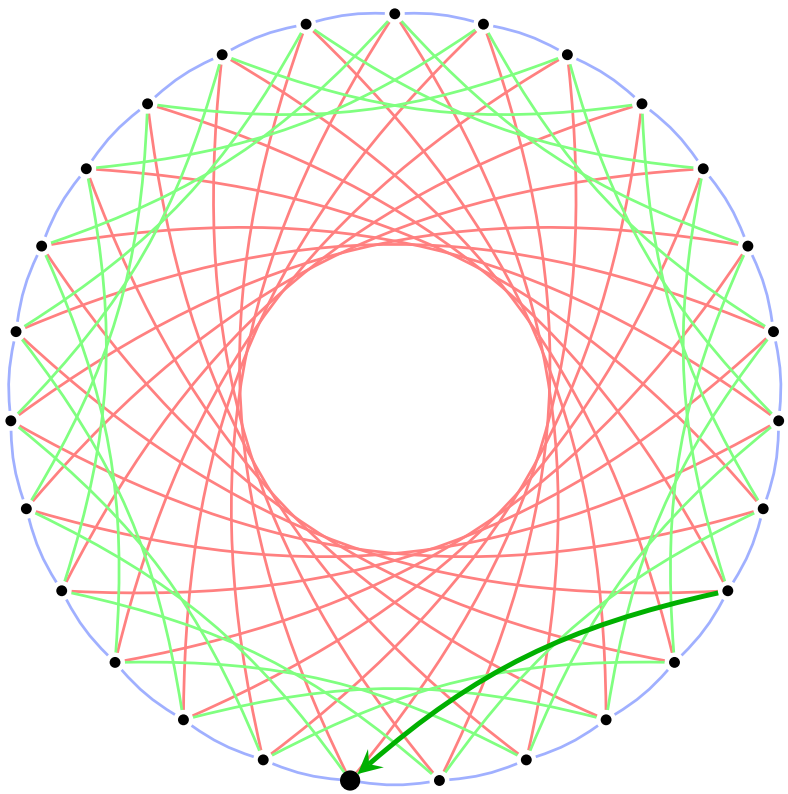
$[-, +, -, -]$



CSIDH key exchange

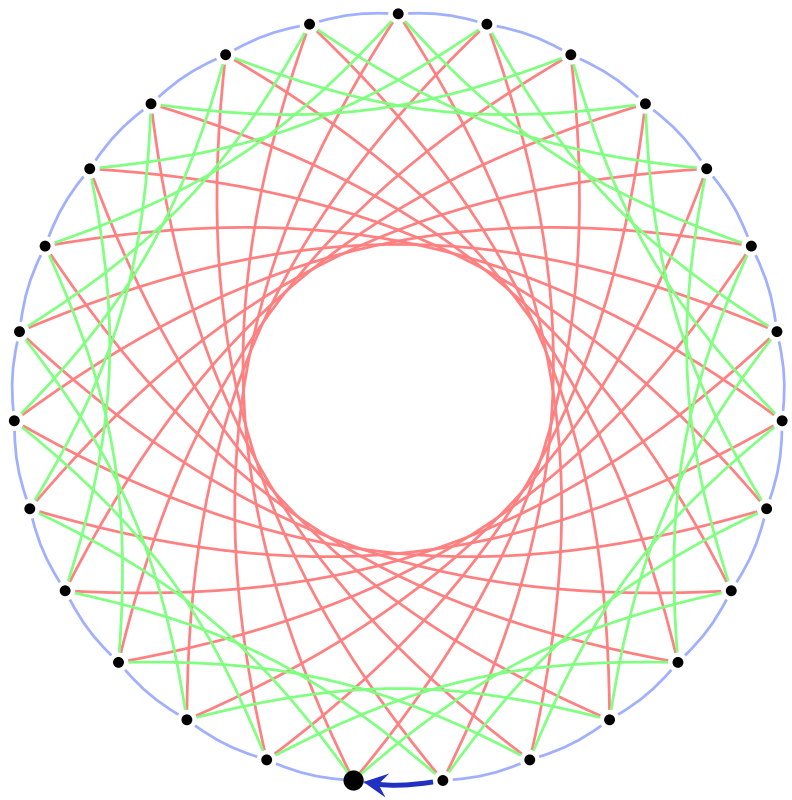
Alice

$[+, +, -, -]$



Bob

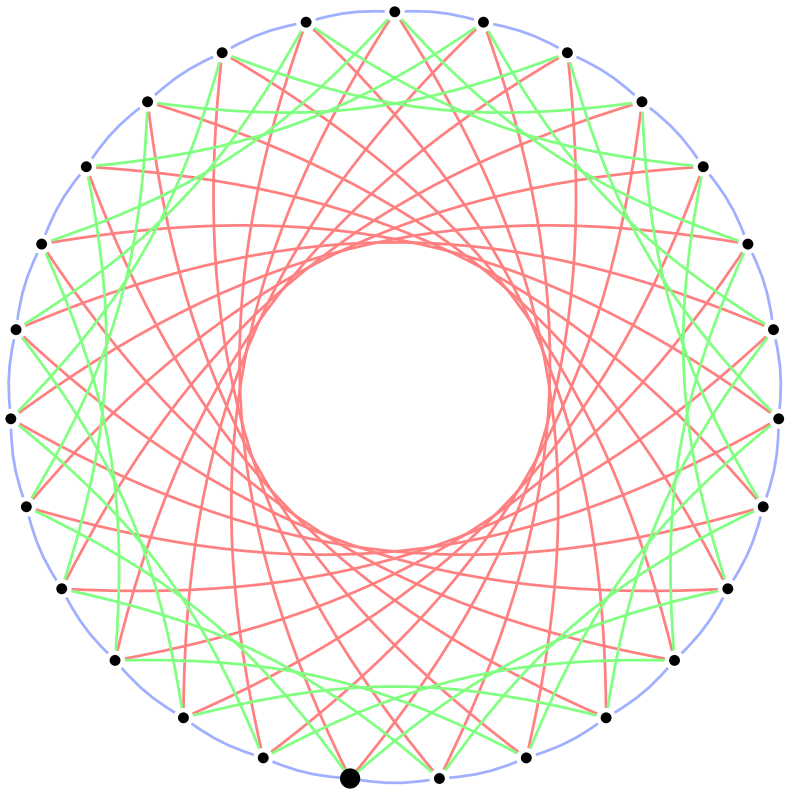
$[-, +, -, -]$



CSIDH key exchange

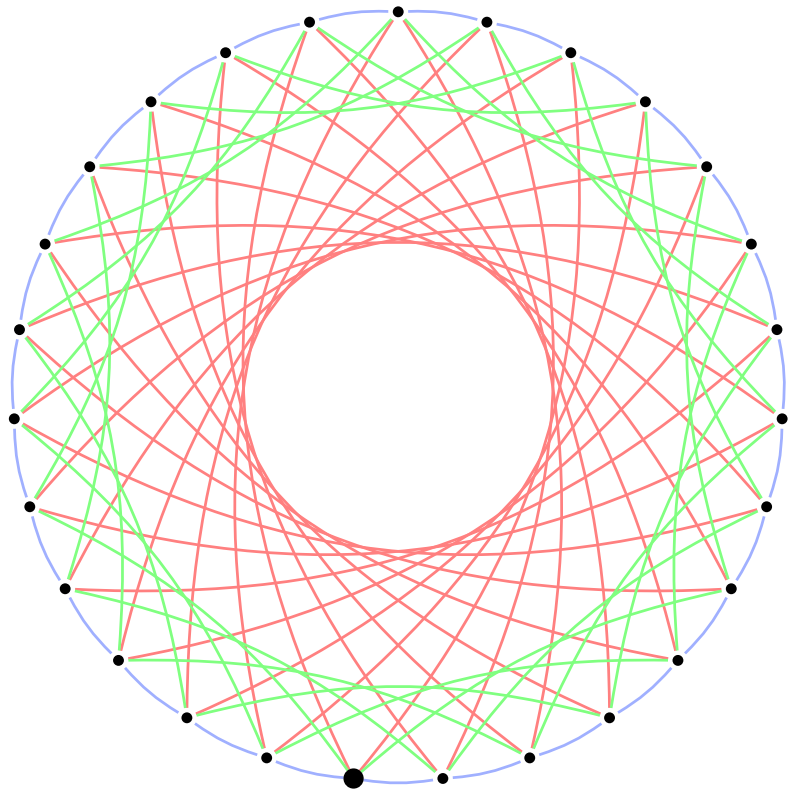
Alice

$[+, +, -, -]$



Bob

$[-, +, -, -]$



Action! 

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

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

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!

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 relation lattice.

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 relation lattice.

!! 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 $\underline{v} \in \mathbb{Z}^n \backslash \{0\}$ which act trivially.
Such $\underline{v}$ form a full-rank subgroup $\Lambda \subseteq \mathbb{Z}^n$, the relation lattice.

!! 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*.

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 relation lattice.

!! 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*.

The lattice Λ is computable in subexponential time classically, and in polynomial time using a quantum computer.
It is used to construct more advanced schemes (*"CSI-FiSh"*).

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

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

For group actions, we simply cannot compose $a * s$ and $b * s$!

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a smooth-degree isogeny $E \to E'$.

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a smooth-degree isogeny $E \to E'$.

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 smooth-degree isogeny $E \to E'$.

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 & friends vary wildly. 

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt[4]{p})$; attacks are basically brute force.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt[4]{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\left((\log p)^{1/2+o(1)}\right)$ due to Kuperberg's quantum algorithm.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt[4]{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security.
(For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt[4]{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security. (For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

- Performance: Some tens of milliseconds per group-action evaluation at the 128-bit *classical* security level.

CSIDH: Where things stand

- Classical security: $\widetilde{O}(\sqrt[4]{p})$; attacks are basically brute force.
- Quantum security: Asymptotically $\exp\big((\log p)^{1/2+o(1)}\big)$ due to Kuperberg's quantum algorithm.

$\Longrightarrow$ Key sizes: Public keys are 4λ bits for *classical* λ-bit security.
(For λ-bit *quantum* security, need $\Theta(\lambda^2)$ bits.)

- Performance: Some tens of milliseconds per group-action evaluation at the 128-bit *classical* security level.

- 2023: "Clapoti" — a polynomial-time algorithm for arbitrary combinations of operations in the group and evaluations of the action. $\rightsquigarrow$ *"KLaPoTi"*, *"PEGASIS"*.
(Previously, only restricted sequences of operations were efficient.)

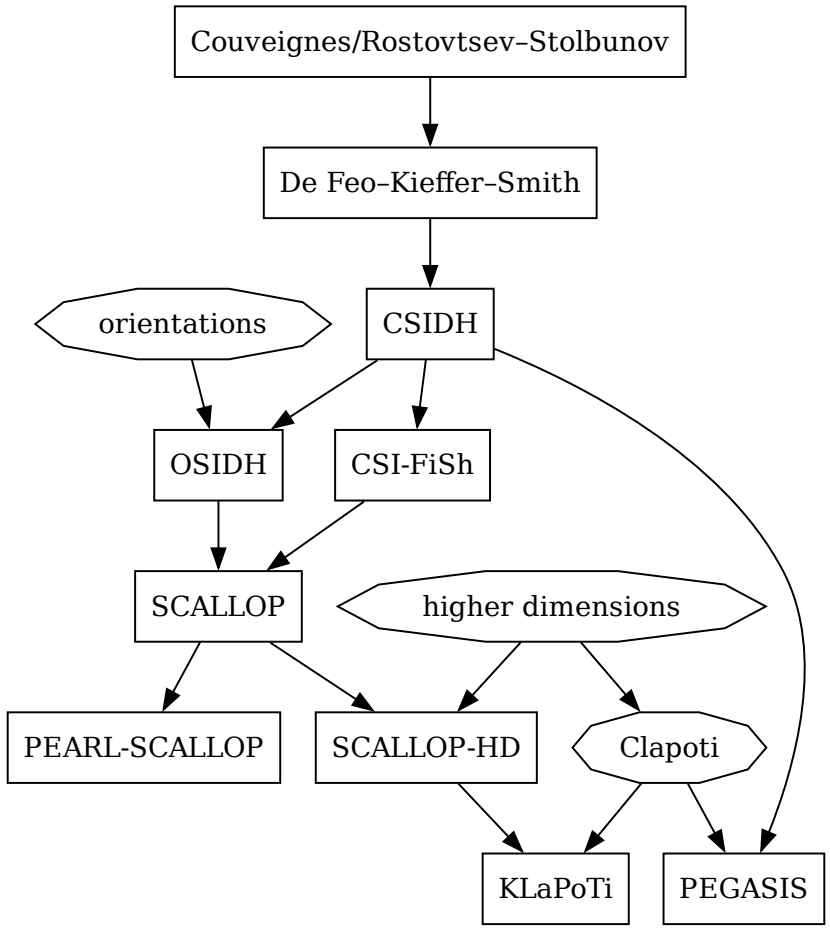
$\rightsquigarrow$ *Friday.* ☺

Other isogeny group actions

There are many ways of building isogeny group actions.

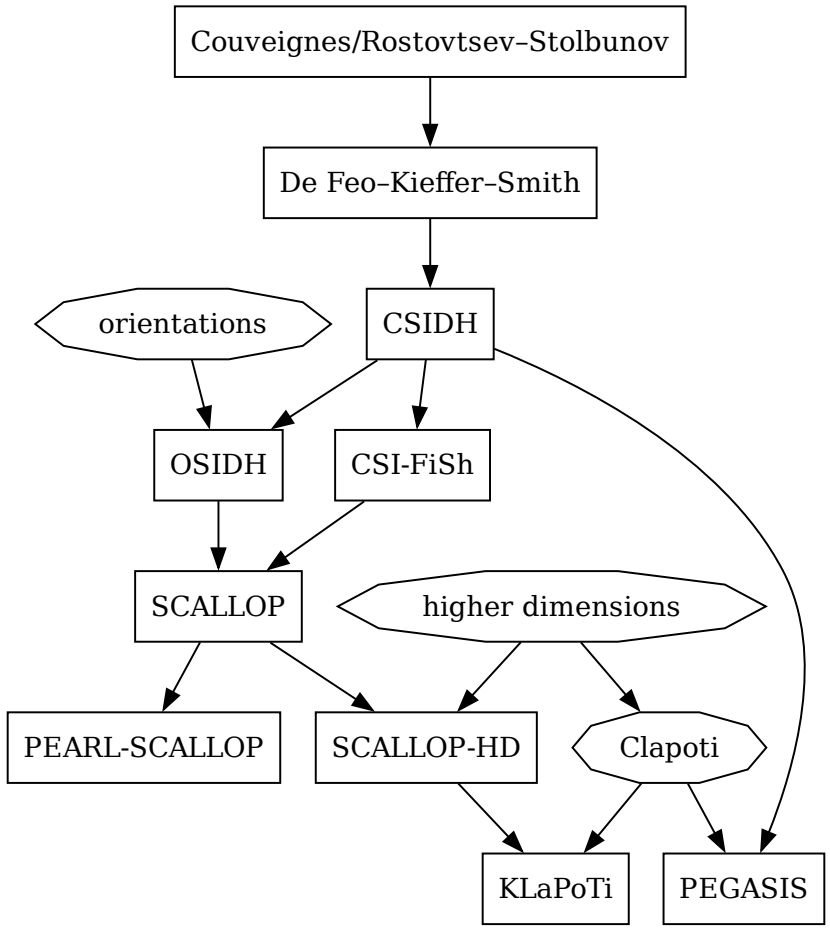
Other isogeny group actions

There are many ways of building isogeny group actions.



Other isogeny group actions

There are many ways of building isogeny group actions.



$\rightsquigarrow$ *Friday.* ☺

Plan for this talk

- Some high-level intuition. ✓
- Elliptic curves & isogenies. ✓
- Isogeny group actions. ✓
- Signatures from isogenies.

SQIsign: What?

https://sqisign.org

SQIsign: What?

https://sqisign.org

- A new-ish and very hot post-quantum signature scheme.
- Based on super cool mathematics. ◡̈

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: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is the Frobenius endomorphism $\pi : (x, y) \mapsto (x^p, y^p)$.)

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is 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: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is 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 an ideal $I_\varphi \subseteq \mathrm{End}(E)$.

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is 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 an ideal $I_\varphi \subseteq \mathrm{End}(E)$.

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

More "special" isogenies

- Earlier: "Special" isogenies φ_ℓ with rational kernel points.
- In other words: $\ker \varphi_\ell = \ker[\ell] \cap \ker(\pi - 1)$.
  (Here π is 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 an ideal $I_\varphi \subseteq \mathrm{End}(E)$.

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

☺ We understand how I_φ, I_ψ relate for isogenies $\varphi, \psi\colon E \to E'$.

$\rightsquigarrow$ one-sided ideal class *set* of $\mathrm{End}(E)$, etc.

The Deuring correspondence

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

The Deuring correspondence

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

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules
  with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

The Deuring correspondence

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

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

A strong connection between two *a priori* very different worlds:

The Deuring correspondence

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

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

a priori

A strong connection between two very different worlds:

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

The Deuring correspondence

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

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

a priori

A strong connection between two 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.

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

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.

The Deuring correspondence

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

Theorem. Fix E_0 supersingular. The (contravariant) functor

$$E \longmapsto \mathrm{Hom}(E, E_0)$$

defines an *equivalence of categories* between

- supersingular elliptic curves with isogenies; and
- invertible left $\mathrm{End}(E_0)$-modules with nonzero left $\mathrm{End}(E_0)$-module homomorphisms.

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.

☺ 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 $E_0 : \; y^2 = x^3 + x$.

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 $E_1 : \; y^2 = x^3 + 1$.

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 $4947 = 3 \cdot 17 \cdot 97$.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- $\approx$All isogeny security reduces to the "$\Longrightarrow$" direction.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- ≈All isogeny security reduces to the "$\Longrightarrow$" direction.
- **SQIsign** builds on the "$\Longleftarrow$" direction constructively.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- ≈All isogeny security reduces to the "$\Longrightarrow$" direction.
- **SQIsign** builds on the "$\Longleftarrow$" direction constructively.
- Essential tool for *both* constructions and attacks.

The Deuring correspondence: Why?

We now know that **the Deuring correspondence lies at the heart of contemporary isogeny-based cryptography.**

(Wesolowski '21: "Orientations and the supersingular endomorphism ring problem").

- $\approx$All isogeny security reduces to the "$\Longrightarrow$" direction.
- **SQIsign** builds on the "$\Longleftarrow$" direction constructively.
- Essential tool for *both* constructions and attacks.

Constructively, *partially* known endomorphism rings are useful.
$\rightsquigarrow$ (**Oriented curves** and) **isogeny group actions**. 🏝️

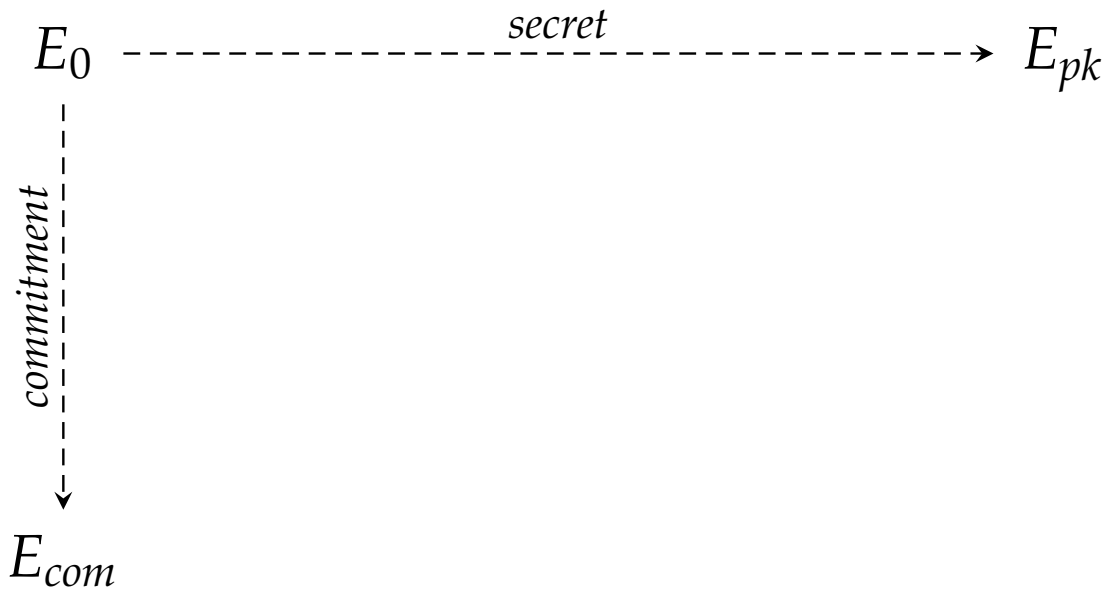
Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.

$$E_0 \xrightarrow{\quad secret \quad} E_{pk}$$

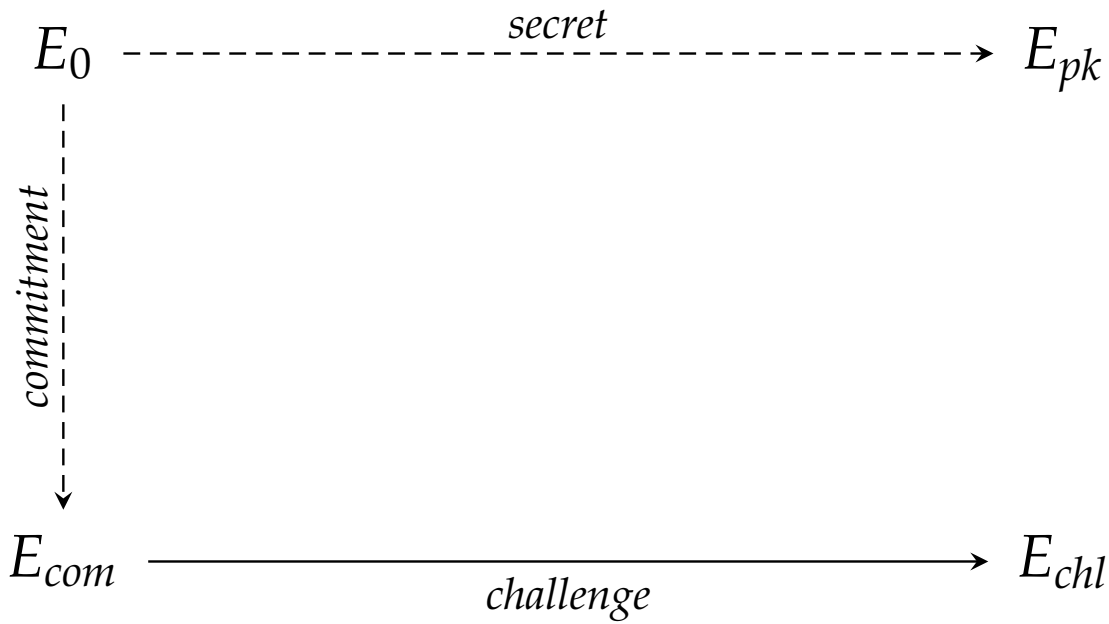
Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.



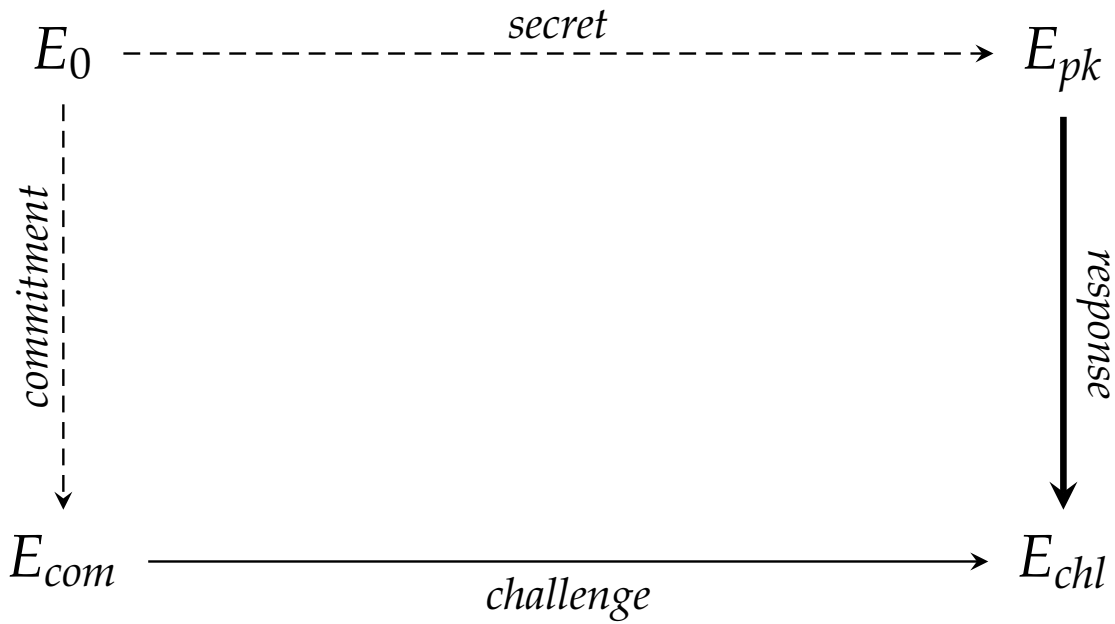
Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.



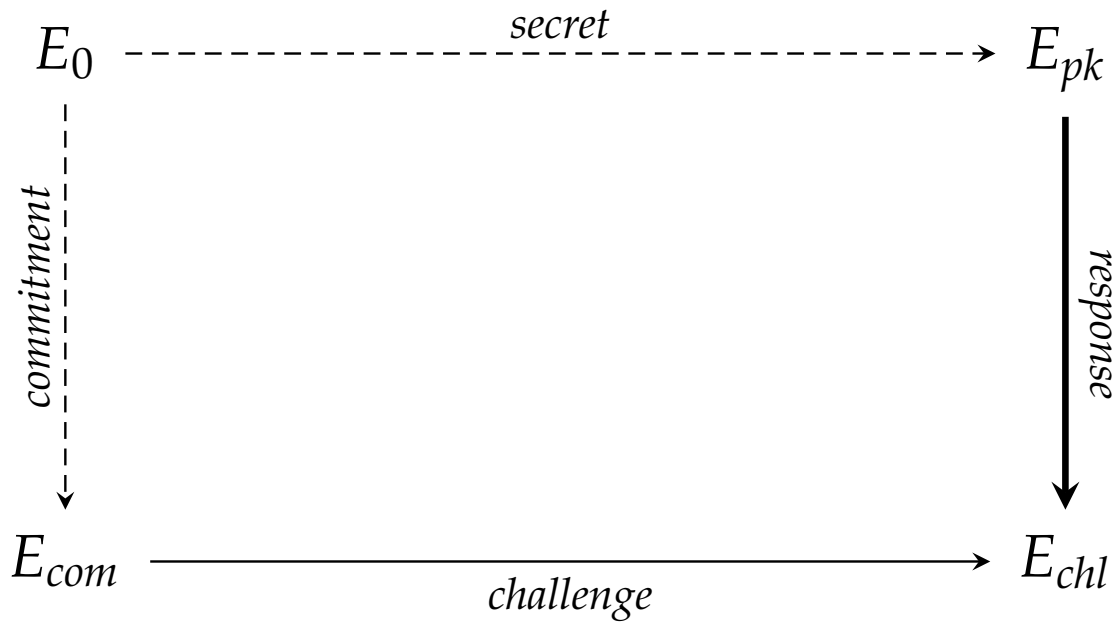
Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.



Signing with isogenies à la SQIsign

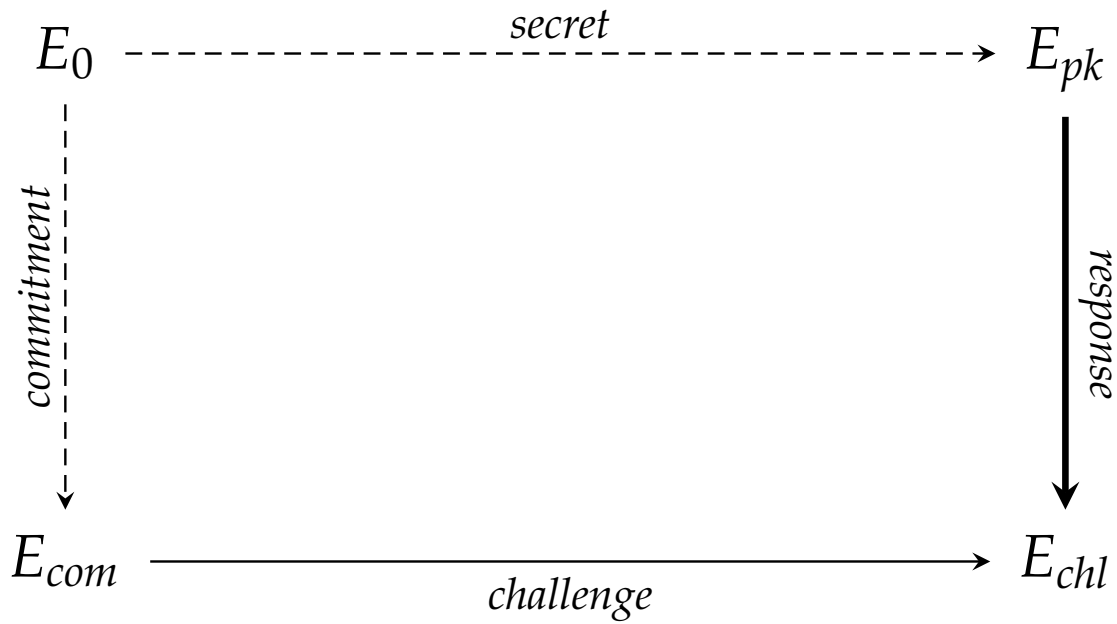
- Fiat–Shamir: signature scheme from identification scheme.



- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*

Signing with isogenies à la SQIsign

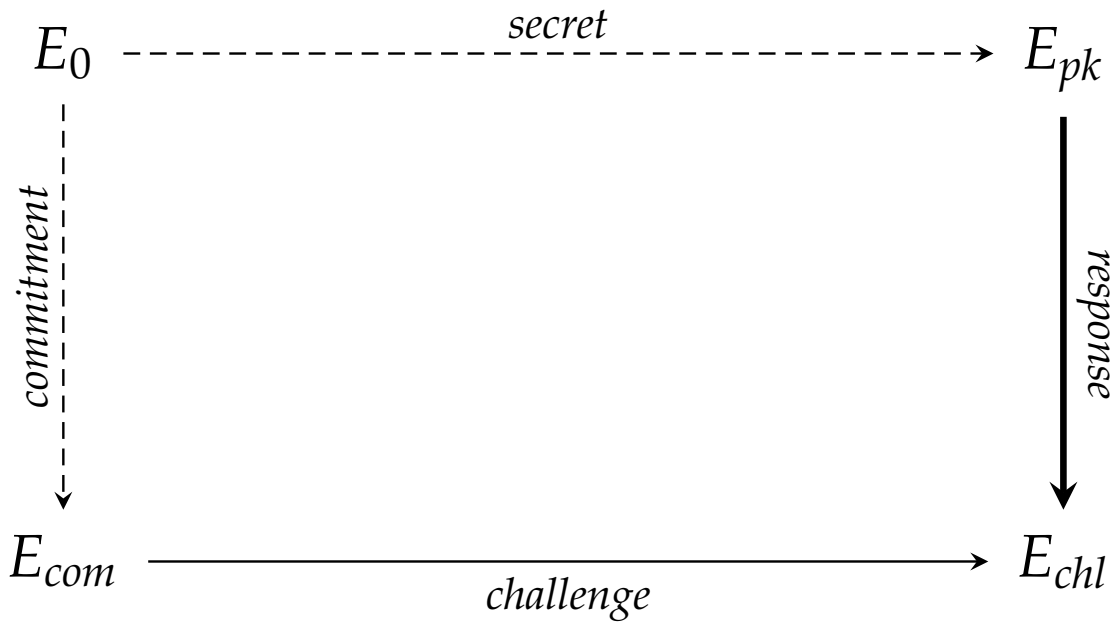
- Fiat–Shamir: signature scheme from identification scheme.



- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*
- **SQIsign**: Construct new path $E_{pk} \to E_{chl}$ (using *secret*).

Signing with isogenies à la SQIsign

- Fiat–Shamir: signature scheme from identification scheme.



- Easy signature: $E_{pk} \to E_0 \to E_{com} \to E_{chl}$. *Obviously broken.*
- **SQIsign**: Construct new path $E_{pk} \to E_{chl}$ (using *secret*).
- It relies on an explicit form of the Deuring correspondence.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

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

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

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

$\rightsquigarrow$ SQIsign rewrites the "broken" signature

$$E_{pk} \to E_0 \to E_{com} \to E_{chl}$$

into a random (smooth) isogeny $E_{pk} \to E_{chl}$.

SQIsign (original version)

Via the Deuring correspondence:

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$.

Main technical tool: The KLPT algorithm.

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

$\rightsquigarrow$ SQIsign rewrites the "broken" signature

$$E_{pk} \to E_0 \to E_{com} \to E_{chl}$$

into a random (smooth) isogeny $E_{pk} \to E_{chl}$.

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

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In SQIsign: Having responses to two distinct challenges means you have a (nontrivial) endomorphism of E_{pk}.

  (Recall that finding endomorphisms is supposedly hard.)

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In SQIsign: Having responses to two distinct challenges means you have a (nontrivial) endomorphism of E_{pk}.

  (Recall that finding endomorphisms is supposedly hard.)

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In SQIsign: Having responses to two distinct challenges means you have a (nontrivial) endomorphism of E_{pk}.

  (Recall that finding endomorphisms is supposedly hard.)

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

  In SQIsign: Somewhat *ad-hoc & heuristic* arguments that "KLPT output is a random isogeny" (from some restricted set).

  ☹ This is not far from the trivial hardness assumption "the scheme is secure"...

SQIsign (original version): Security

For SQIsign to be secure, we require two main properties:

- Soundness: Creating valid signatures requires either knowing the private key or solving a hard problem.

  In SQIsign: Having responses to two distinct challenges means you have a (nontrivial) endomorphism of E_{pk}.

  (Recall that finding endomorphisms is supposedly hard.)

- Zero-knowledge: Signatures are (very close to) statistically independent of the secret key.

  In SQIsign: Somewhat *ad-hoc & heuristic* arguments that "KLPT output is a random isogeny" (from some restricted set).

  ☹ This is not far from the trivial hardness assumption "the scheme is secure"...

Known attacks for endomorphism-ring problem:
$O(\sqrt{p})$ classically, $O(\sqrt[4]{p})$ quantumly. Fully exponential! 🎉

SQIsign: Why?

+ It's extremely small compared to the competition.
– It's relatively slow compared to the competition.
+ ...but performance only gets better!

SQIsign: Why?

+ It's extremely small compared to the competition.
– It's relatively slow compared to the competition.
+ ...but performance only gets better!

SQIsign (original version): 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` (2023–2024)

☺ Timings have gotten *much* better since. $\rightsquigarrow$ *Friday.* ☺

Plan for this talk

- Some high-level intuition. ✓
- Elliptic curves & isogenies. ✓
- Isogeny group actions. ✓
- Signatures from isogenies. ✓

Ad break

https://isogeny.club

Questions?

lorenz@yx7.cc