Elliptic-curve and isogeny-based cryptography

Chloe Martindale

University of Bristol

SLMath Summer School
Isogeny-based cryptography
Day 2

Recall: Diffie–Hellman key exchange ’76

Public parameters:

- a prime p (experts: uses $\mathbb{F}_p^*$, today also elliptic curves)
- a number $g \pmod{p}$ (nonexperts: think of an integer less than p)

Alice	Eve	Bob
$\mathrm{sk}_A \xleftarrow{\text{random}} \{0 \ldots p-1\}$		$\mathrm{sk}_B \xleftarrow{\text{random}} \{0 \ldots p-1\}$
$\mathrm{pk}_A = g^{\mathrm{sk}_A}$		$\mathrm{pk}_B = g^{\mathrm{sk}_B}$
$ss := (g^{\mathrm{sk}_B})^{\mathrm{sk}_A}$		$ss := (g^{\mathrm{sk}_A})^{\mathrm{sk}_B}$

- Alice and Bob agree on a shared secret key ss, then they can use that to encrypt their messages.
- Eve sees $\mathrm{pk}_A = g^{\mathrm{sk}_A}$, $\mathrm{pk}_B = g^{\mathrm{sk}_B}$; can’t find sk_A, sk_B, ss.

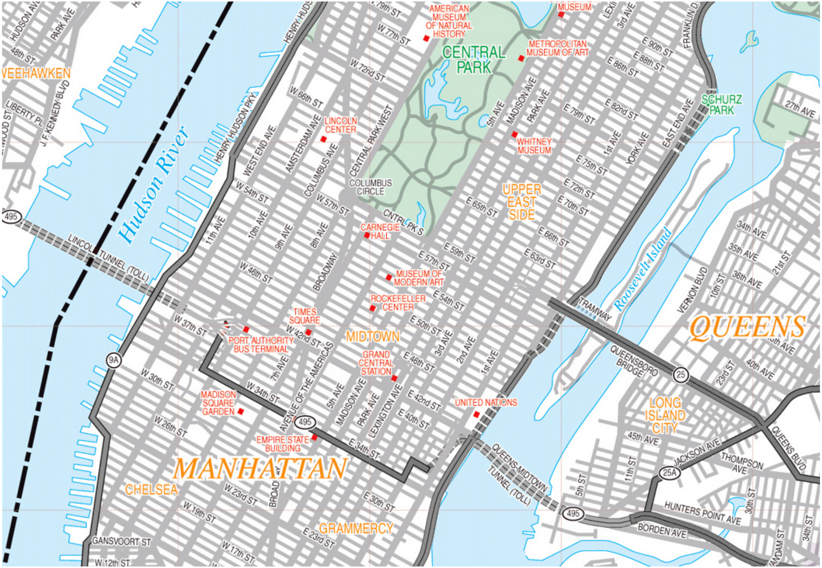
Recall: Diffie–Hellman key exchange '76

Public parameters:

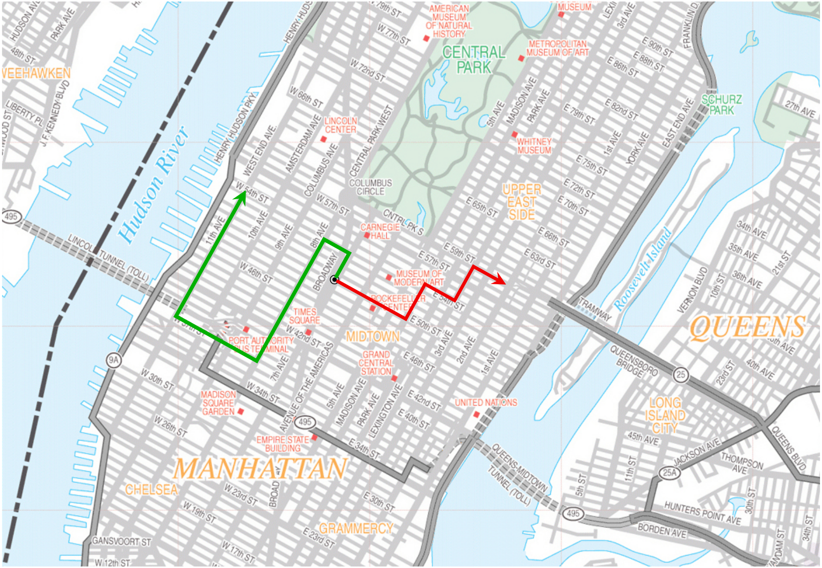
- a prime p (experts: uses $\mathbb{F}_p^*$, today also elliptic curves)
- a number $g \pmod{p}$ (nonexperts: think of an integer less than p)

- Alice and Bob agree on a shared secret key ss, then they can use that to encrypt their messages.
- Eve sees $\mathrm{pk}_A = g^{\mathrm{sk}_A}$, $\mathrm{pk}_B = g^{\mathrm{sk}_B}$; can't find sk_A, sk_B, ss.

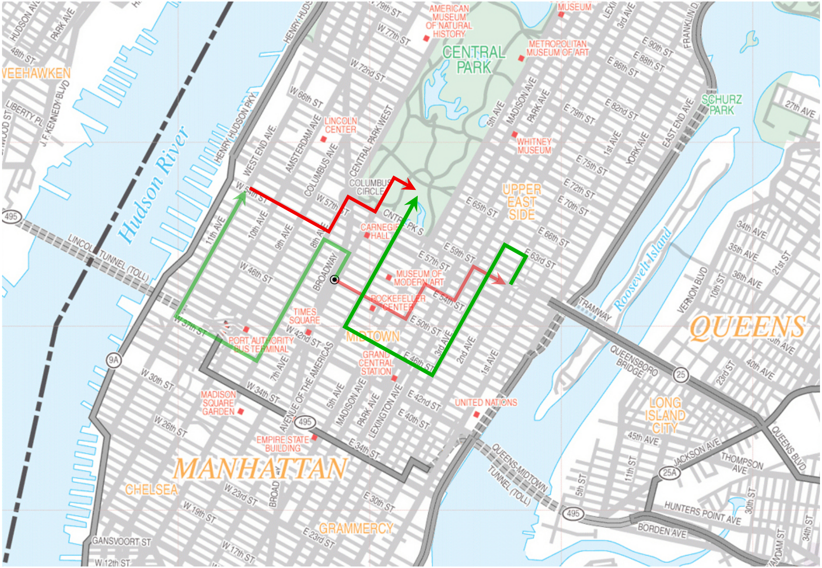
Graph walking Diffie–Hellman?



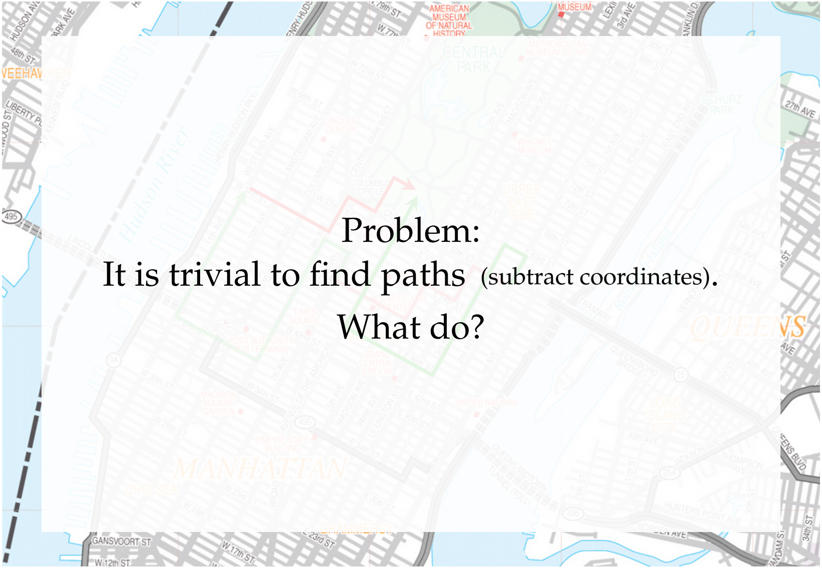
Graph walking Diffie–Hellman?



Graph walking Diffie–Hellman?



Graph walking Diffie–Hellman?



Problem:

It is trivial to find paths (subtract coordinates).

What do?

Big picture

- Isogenies are a source of exponentially-sized graphs.

Big picture

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

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.
- Enough structure to navigate the graph meaningfully.
  That is: some *well-behaved* ‘directions’ to describe paths. More later.

Big picture

- Isogenies are a source of exponentially-sized graphs.
- We can walk efficiently on these graphs.
- Fast mixing: short paths to (almost) all nodes.
- No known efficient algorithms to recover paths from endpoints.
- Enough structure to navigate the graph meaningfully.
  That is: some *well-behaved* 'directions' to describe paths. More later.

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

Stand back!

We’re going to do maths.

Maths background #1/3: Isogenies *(edges)*

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

- given by rational functions.
- a group homomorphism.

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

Maths background #1/3: Isogenies *(edges)*

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

- given by rational functions.
- a group homomorphism.

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

An endomorphism of E is an isogeny $E \to E$, or the zero map.
The ring of endomorphisms of E is denoted by $\mathrm{End}(E)$.

Maths background #1/3: Isogenies *(edges)*

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

- given by rational functions.
- a group homomorphism.

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

An endomorphism of E is an isogeny $E \to E$, or the zero map.
The ring of endomorphisms of E is denoted by $\mathrm{End}(E)$.

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

Maths background #2/3: 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')

Maths background #2/3: 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.

Vélu '71:
Formulas for computing E/G and evaluating φ_G at a point.

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

[1](up to isomorphism of E')

Maths background #2/3: 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.

Vélu '71:
Formulas for computing E/G and evaluating φ_G at a point.

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

Vélu operates in the field where the points in G live.

$\rightsquigarrow$ need to make sure extensions stay small for desired $\#G$

$\rightsquigarrow$ this is why we use supersingular curves!

[1](up to isomorphism of E')

Math slide #3/3: Supersingular isogeny graphs

Let p be a prime, q a power of p, and ℓ a positive integer $\notin p\mathbb{Z}$.

An elliptic curve $E/\mathbb{F}_q$ is *supersingular* if $p \mid (q + 1 - \#E(\mathbb{F}_q))$.

We care about the cases $\#E(\mathbb{F}_p) = p + 1$ and $\#E(\mathbb{F}_{p^2}) = (p + 1)^2$.

$\rightsquigarrow$ easy way to control the group structure by choosing p!

Math slide #3/3: Supersingular isogeny graphs

Let p be a prime, q a power of p, and ℓ a positive integer $\notin p\mathbb{Z}$.

An elliptic curve $E/\mathbb{F}_q$ is *supersingular* if $p \mid (q + 1 - \#E(\mathbb{F}_q))$.

We care about the cases $\#E(\mathbb{F}_p) = p + 1$ and $\#E(\mathbb{F}_{p^2}) = (p + 1)^2$.

$\rightsquigarrow$ easy way to control the group structure by choosing p!

Let $S \not\ni p$ denote a set of prime numbers.

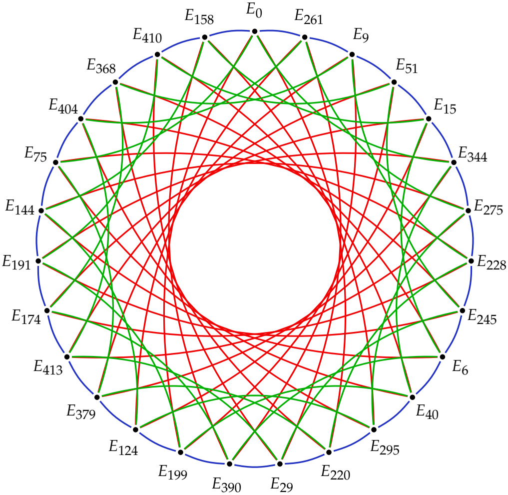
The supersingular S-isogeny graph over $\mathbb{F}_q$ consists of:

- vertices given by isomorphism classes of supersingular elliptic curves,
- edges given by equivalence classes[1] of ℓ-isogenies ($\ell \in S$),

both defined over $\mathbb{F}_q$.

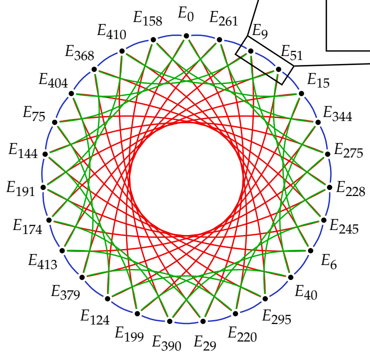
[1] Two isogenies $\varphi\colon E \to E'$ and $\psi\colon E \to E''$ are identified if $\psi = \iota \circ \varphi$ for some isomorphism $\iota\colon E' \to E''$.

Graphs of elliptic curves

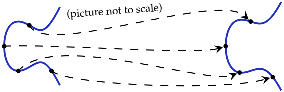


Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Edges: 3-, 5-, and 7-isogenies

Graphs of elliptic curves



A 3-isogeny



$$E_{51}\colon y^2 = x^3 + 51x^2 + x \longrightarrow E_9\colon y^2 = x^3 + 9x^2 + x$$

$$(x, y) \longmapsto \left(\frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97},\; y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)$$

[ˈsiːˌsaɪd]

CRS or CSIDH

Traditionally, Diffie-Hellman works in a group G via the map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

CRS or CSIDH

Traditionally, Diffie-Hellman works in a group G via the map

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Shor's algorithm quantumly computes x from g^x in any group in polynomial time.

CRS or CSIDH

Traditionally, Diffie-Hellman works in a group G via the map

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Shor's algorithm quantumly computes x from g^x in any group in polynomial time.

$\rightsquigarrow$ Idea:

Replace exponentiation on the group G by a group action of a group H on a set S:

$$H \times S \to S.$$

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

by a group action on a set.

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

by a group action on a set.

- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{rcl} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.
- Replace $\mathbb{Z}$ by the commutative group $\mathrm{cl}(\mathbb{Z}\sqrt{-p}])$.

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.
- Replace $\mathbb{Z}$ by the commutative group $\mathrm{cl}(\mathbb{Z}\sqrt{-p}])$.
- An ideal in $\mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E_A))$ is the kernel of an isogeny from E_A.

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.
- Replace $\mathbb{Z}$ by the commutative group $\mathrm{cl}(\mathbb{Z}\sqrt{-p}])$.
- An ideal in $\mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E_A))$ is the kernel of an isogeny from E_A.
- The action of a well-chosen $\mathfrak{l} \in \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ on S moves the elliptic curves one step around one of the cycles.

$$\begin{array}{ccc} \mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) \times S & \to & S \\ (\mathfrak{l}_3, E) & \mapsto & \mathfrak{l}_3 * E. \end{array}$$

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.
- Replace $\mathbb{Z}$ by the commutative group $\mathrm{cl}(\mathbb{Z}\sqrt{-p}])$.
- An ideal in $\mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E_A))$ is the kernel of an isogeny from E_A.
- The action of a well-chosen $\mathfrak{l} \in \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ on S moves the elliptic curves one step around one of the cycles.

$$\begin{array}{ccc} \mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) \times S & \to & S \\ (\mathfrak{l}_5, E) & \mapsto & \mathfrak{l}_5 * E. \end{array}$$

Quantumifying Exponentiation

- We want to replace the exponentiation map

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x \end{array}$$

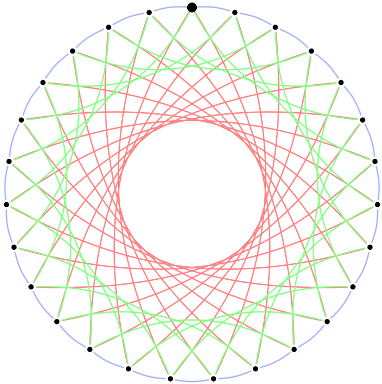
  by a group action on a set.
- Replace G by the set S of supersingular elliptic curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
- For every $E_A \in S$, the ring of $\mathbb{F}_p$-rational endomorphisms $\mathrm{End}_{\mathbb{F}_p}(E_A)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$.
- Replace $\mathbb{Z}$ by the commutative group $\mathrm{cl}(\mathbb{Z}\sqrt{-p}])$.
- An ideal in $\mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E_A))$ is the kernel of an isogeny from E_A.
- The action of a well-chosen $\mathfrak{l} \in \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ on S moves the elliptic curves one step around one of the cycles.

$$\begin{array}{ccc} \mathrm{cl}(\mathbb{Z}[\sqrt{-p}]) \times S & \to & S \\ (\mathfrak{l}_7, E) & \mapsto & \mathfrak{l}_7 * E. \end{array}$$

Diffie and Hellman go to the CSIDH

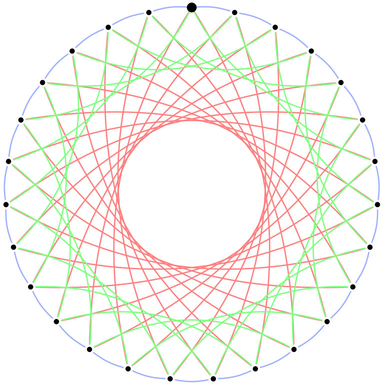
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$



Diffie and Hellman go to the CSIDH

Alice

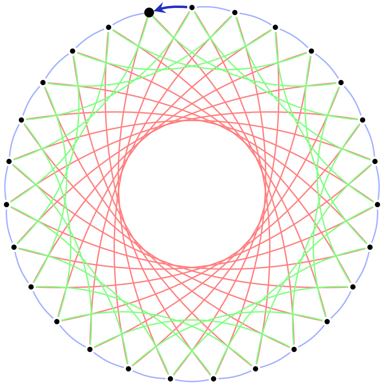
$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$

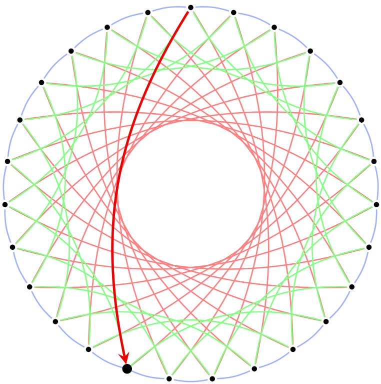
$\uparrow$

Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$

$\uparrow$





Diffie and Hellman go to the CSIDH

Alice

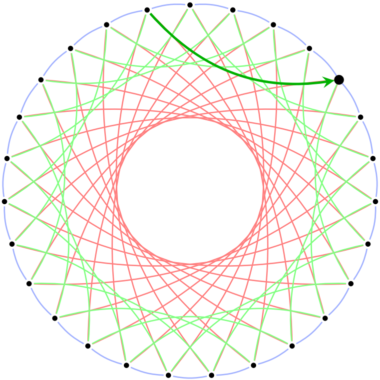
$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$

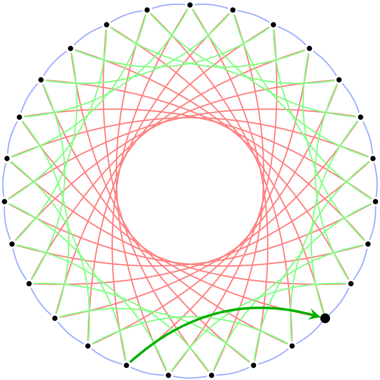
$\uparrow$

Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$

$\uparrow$

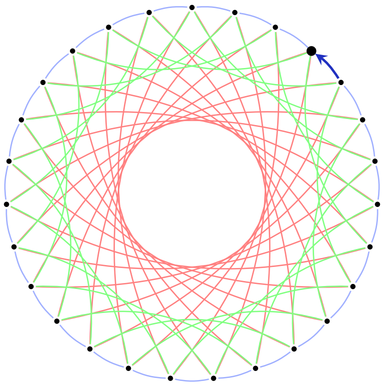




Diffie and Hellman go to the CSIDH

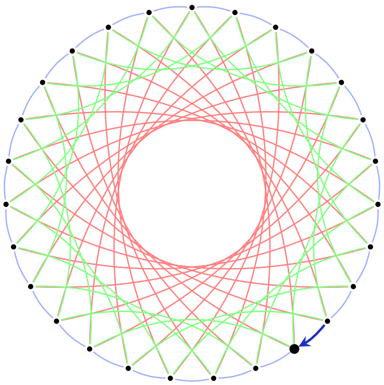
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \underset{\uparrow}{\mathfrak{l}_3}, \mathfrak{l}_5{}^{-1}]$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \underset{\uparrow}{\mathfrak{l}_3{}^{-1}}, \mathfrak{l}_5]$



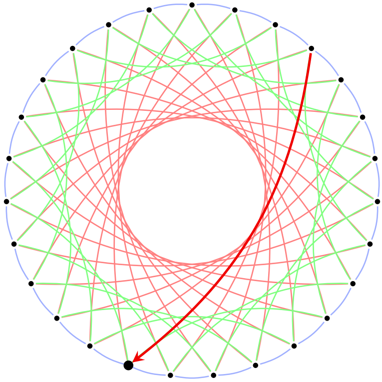
Diffie and Hellman go to the CSIDH

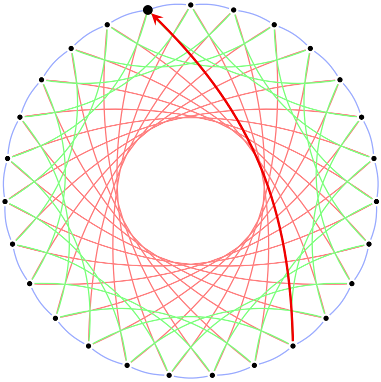
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \underset{\uparrow}{\mathfrak{l}_5{}^{-1}}]$

Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \underset{\uparrow}{\mathfrak{l}_5}]$





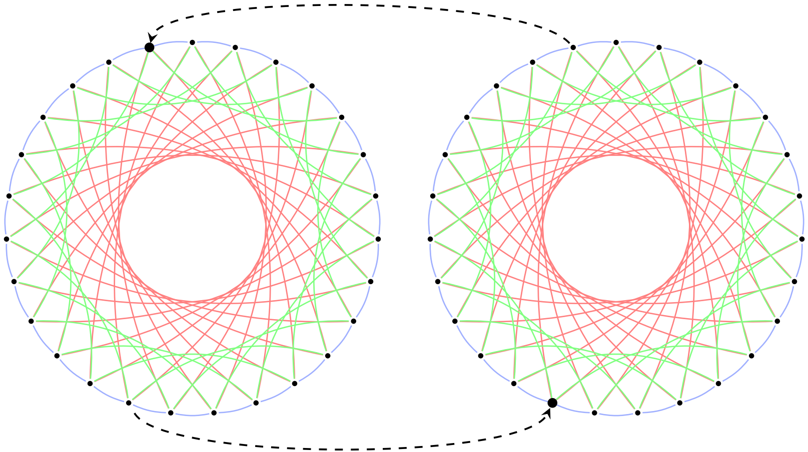
Diffie and Hellman go to the CSIDH

Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$

Bob

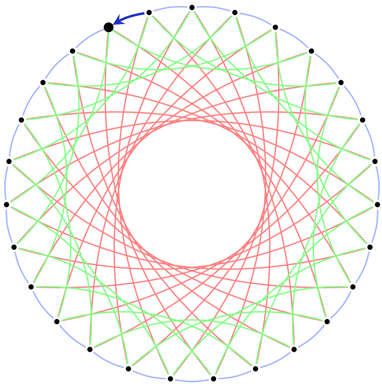
$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$



Diffie and Hellman go to the CSIDH

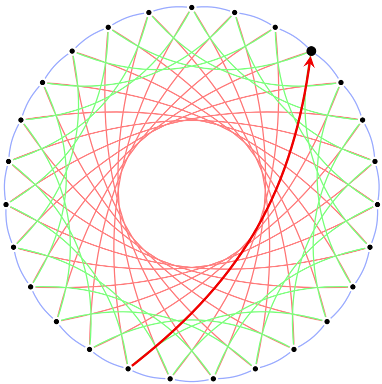
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$
$\uparrow$



Bob

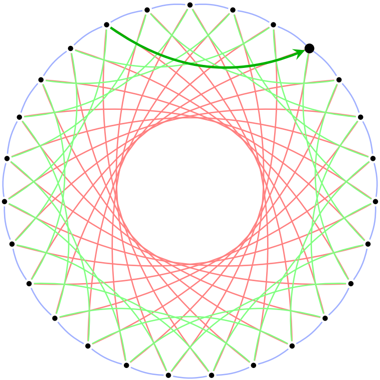
$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$
$\uparrow$



Diffie and Hellman go to the CSIDH

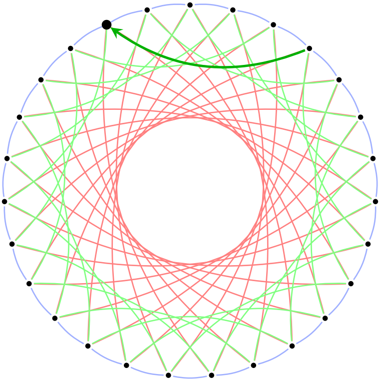
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$

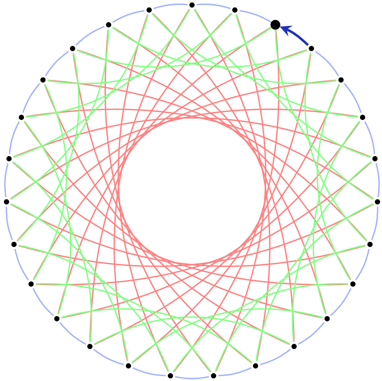


Diffie and Hellman go to the CSIDH

Alice

$[\mathfrak{l}_3, \mathfrak{l}_7^{-1}, \mathfrak{l}_3, \mathfrak{l}_5^{-1}]$

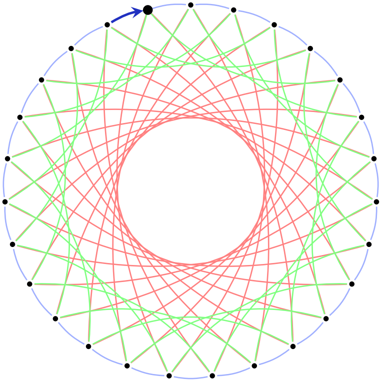
$\uparrow$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3^{-1}, \mathfrak{l}_5]$

$\uparrow$

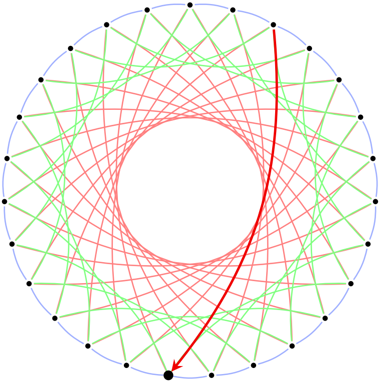


Diffie and Hellman go to the CSIDH

Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$

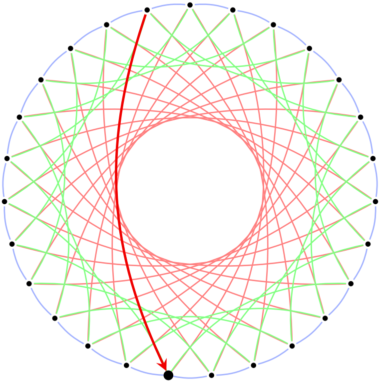
$\uparrow$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$

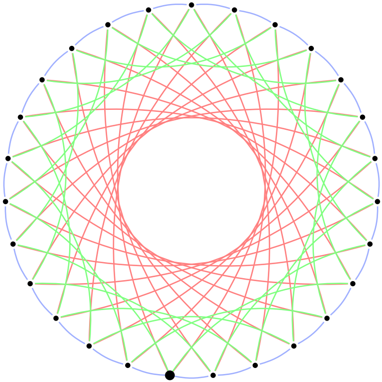
$\uparrow$



Diffie and Hellman go to the CSIDH

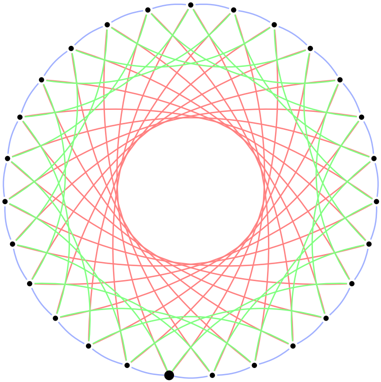
Alice

$[\mathfrak{l}_3, \mathfrak{l}_7{}^{-1}, \mathfrak{l}_3, \mathfrak{l}_5{}^{-1}]$



Bob

$[\mathfrak{l}_5, \mathfrak{l}_7, \mathfrak{l}_3{}^{-1}, \mathfrak{l}_5]$



Choosing parameters

In [CLMPR18], parameters are chosen as follows:

- $\ell_1, \ldots, \ell_{n-1}$ the first $n-1$ odd primes.
- $\ell_n > \ell_{n-1}$ the smallest prime such that $p = 4\ell_1 \cdots \ell_n - 1$ is prime.

Then:

- $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ correspond to kernels of $\mathbb{F}_p$-rational isogenies (see next slide) — fast.
- Allowing up to 5 actions of each $\mathfrak{l}_i^{(-1)}$ covers* the whole class group — security then depends on size of class group.

Choosing parameters

In [CLMPR18], parameters are chosen as follows:

- $\ell_1, \ldots, \ell_{n-1}$ the first $n-1$ odd primes.
- $\ell_n > \ell_{n-1}$ the smallest prime such that $p = 4\ell_1 \cdots \ell_n - 1$ is prime.

Then:

- $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ correspond to kernels of $\mathbb{F}_p$-rational isogenies (see next slide) — fast.
- Allowing up to 5 actions of each $\mathfrak{l}_i^{(-1)}$ covers* the whole class group — security then depends on size of class group.

*Any $I \in \mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ can be written as $\prod \mathfrak{l}_i^{e_i}$ with $e_i \in [-5, \ldots, 5]$.

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.

- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p+1$ or $(p+1)/2$.

- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p+1$ or $(p+1)/2$.
  - For every odd prime $\ell | (p+1)$, the point $\frac{p+1}{\ell}P$ is a point of order ℓ.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).

Compute neighbours in the graph

To compute a neighbour of E, we have to compute an ℓ-isogeny from E. To do this:

- Find a point P of order ℓ on E.
  - Let $E/\mathbb{F}_p$ be supersingular and $p \geq 5$. Then $E(\mathbb{F}_p) \cong C_{p+1}$ or $C_2 \times C_{(p+1)/2}$.
  - Suppose we have found $P = E(\mathbb{F}_p)$ of order $p+1$ or $(p+1)/2$.
  - For every odd prime $\ell | (p+1)$, the point $\frac{p+1}{\ell}P$ is a point of order ℓ.
- Compute the isogeny with kernel $\{P, 2P, \ldots, \ell P\}$ using Vélu's formulas* (implemented in Sage).
  - Given a $\mathbb{F}_p$-rational point of order ℓ, the isogeny computations can be done over $\mathbb{F}_p$.

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A : y^2 = x^3 + Ax^2 + x.$$

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A : y^2 = x^3 + Ax^2 + x.$$

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$.

Representing nodes of the graph

- Every node of G_{ℓ_i} is

$$E_A : y^2 = x^3 + Ax^2 + x.$$

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$.

$\Rightarrow$ Tiny keys!

Does any A work?

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Does any A work?

No.

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p+1$ points.

Does any A work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.

[1]This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Does any A work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.
- Public-key validation: Check that E_A has $p + 1$ points.

  Easy Monte-Carlo algorithm: Pick random P on E_A and check $[p + 1]P = \infty$.[1]

[1] This algorithm has a small chance of false positives, but we actually use a variant that *proves* that E_A has $p + 1$ points.

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

- Kuperberg's algorithm [Kup1] requires a subexponential number of queries, and a subexponential number of operations on a subexponential number of qubits.

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

- Kuperberg's algorithm [Kup1] requires a subexponential number of queries, and a subexponential number of operations on a subexponential number of qubits.
- Variant by Regev [Reg] uses polynomial number of qubits at the expense of time.

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

- Kuperberg's algorithm [Kup1] requires a subexponential number of queries, and a subexponential number of operations on a subexponential number of qubits.
- Variant by Regev [Reg] uses polynomial number of qubits at the expense of time.
- Kuperberg later [Kup2] gave more trade-off options for quantum and classical memory vs. time.

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

- Kuperberg's algorithm [Kup1] requires a subexponential number of queries, and a subexponential number of operations on a subexponential number of qubits.
- Variant by Regev [Reg] uses polynomial number of qubits at the expense of time.
- Kuperberg later [Kup2] gave more trade-off options for quantum and classical memory vs. time.
- Childs-Jao-Soukharev [CJS] applied Kuperberg/Regev to CRS – their attack also applies to CSIDH.

Quantum Security

Hidden-shift algorithms: Subexponential complexity (Kuperberg, Regev).

- Kuperberg's algorithm [Kup1] requires a subexponential number of queries, and a subexponential number of operations on a subexponential number of qubits.
- Variant by Regev [Reg] uses polynomial number of qubits at the expense of time.
- Kuperberg later [Kup2] gave more trade-off options for quantum and classical memory vs. time.
- Childs-Jao-Soukharev [CJS] applied Kuperberg/Regev to CRS – their attack also applies to CSIDH.
- Part of CJS attack computes many paths in superposition.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.
- For fastest variant of Kuperberg, total cost of CSIDH-512 attack is at least 2^{56} qubit operations.

Quantum Security

Original proposal in 2018 paper: $\mathbb{F}_p \approx 512$ bits.

- The exact cost of the Kuperberg/Regev/CJS attack is subtle – it depends on:
  - Choice of time/memory trade-off (Regev/Kuperberg)
  - Quantum evaluation of isogenies

  (and much more).
- [BLMP19] computes one query (i.e. CSIDH-512 group action) using $765325228976 \approx 0.7 \cdot 2^{40}$ nonlinear bit operations.
- Peikert's sieve technique [P19] on fastest variant of Kuperberg requires 2^{16} queries using 2^{40} bits of quantum accessible classical memory.
- For fastest variant of Kuperberg, total cost of CSIDH-512 attack is at least 2^{56} qubit operations.
- Overheads from error correction, high quantum memory etc., not yet understood.

Better parameters - SQALE

[CCJR22] propose the SQALE of CSIDH.

- Uses huge $p = 4\ell_1 \cdots \ell_n - 1$

Better parameters - SQALE

[CCJR22] propose the SQALE of CSIDH.

- Uses huge $p = 4\ell_1 \cdots \ell_n - 1$
- Uses only $\mathfrak{l}_i^{\pm 1}$

Better parameters - SQALE

[CCJR22] propose the SQALE of CSIDH.

- Uses huge $p = 4\ell_1 \cdots \ell_n - 1$
- Uses only $\mathfrak{l}_i^{\pm 1}$
- Tiny fraction of class group used

Better parameters - SQALE

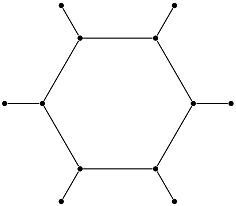
[CCJR22] propose the SQALE of CSIDH.

- Uses huge $p = 4\ell_1 \cdots \ell_n - 1$
- Uses only $\mathfrak{l}_i^{\pm 1}$
- Tiny fraction of class group used
- Not a subgroup $\rightsquigarrow$ Kuperberg has to use huge group

Better parameters - CSURF

Q: What about 2-isogenies?

- The 2-isogeny graph looks like this:

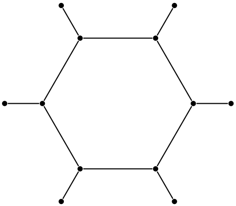


- This is called an isogeny volcano.

Better parameters - CSURF

Q: What about 2-isogenies?

- The 2-isogeny graph looks like this:

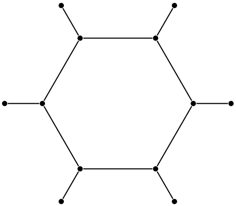


- This is called an isogeny volcano.
- Edges on the cycle are horizontal.

Better parameters - CSURF

Q: What about 2-isogenies?

- The 2-isogeny graph looks like this:

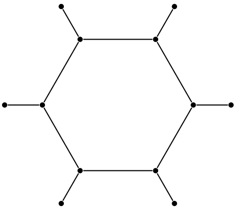


- This is called an isogeny volcano.
- Edges on the cycle are horizontal.
- Away / back to the cycle is descending / ascending.

Better parameters - CSURF

Q: What about 2-isogenies?

- The 2-isogeny graph looks like this:



- This is called an isogeny volcano.
- Edges on the cycle are horizontal.
- Away / back to the cycle is descending / ascending.

$\rightsquigarrow$ How to compute 'on the surface'?

Better parameters - CSURF

[CD19] solve these problems:

- Set $p = 4f\ell_1 \cdots \ell_n - 1$ where $\ell_1 = 2$.

Better parameters - CSURF

[CD19] solve these problems:

- Set $p = 4f\ell_1 \cdots \ell_n - 1$ where $\ell_1 = 2$.
- Set $E_0/\mathbb{F}_p : y^2 = x^3 - x$. Then E_0 is 'on the surface'.

Better parameters - CSURF

[CD19] solve these problems:

- Set $p = 4f\ell_1 \cdots \ell_n - 1$ where $\ell_1 = 2$.
- Set $E_0/\mathbb{F}_p : y^2 = x^3 - x$. Then E_0 is 'on the surface'.
- For any curve on the surface, the 2-isogeny with kernel $\langle(0, 0)\rangle$ is horizontal.

Venturing further beyond the CSIDH

A selection of more advances since original publication (2018):

- sqrtVelu [BDLS20]: square-root speed-up on computation of large-degree isogenies.
- Radical isogenies [CDV20]: significant speed-up on isogenies of small-ish degree.
- Some work on different curve forms (e.g. Edwards).
- Knowledge of $\mathrm{End}(E_0)$ and $\mathrm{End}(E_A)$ breaks CSIDH in classical polynomial time [Wes21].
- CTIDH [$\mathrm{B}^2\mathrm{C}^2\mathrm{LMS}^2$]: Efficient constant-time CSIDH-style construction.

References

[$B^2C^2LMS^2$]	`ctidh.isogeny.org`
[BD17]	`ia.cr/2017/334`
[BDLS20]	`velusqrt.isogeny.org`
[BEG19]	`ia.cr/2019/485`
[BLMP19]	`quantum.isogeny.org`
[CCJR22]	`ia.cr/2020/1520`
[CD19]	`ia.cr/2019/1404`
[CDV20]	`ia.cr/2020/1108`
[FM19]	`ia.cr/2019/555`
[GMT19]	`ia.cr/2019/431`
[Wes21]	`ia.cr/2021/1583`