Introduction to isogeny-based cryptography

Lorenz Panny

Technische Universiteit Eindhoven

AIM, San Jose, 7 February 2019

Words are hard

> "So... How's it going with your isonegies?"
>
> — a lattice-based crypto researcher

Words are hard

“So... How’s it going with your isonegies?”
— a lattice-based crypto researcher

...I mean, a carbon-based researcher who works on lattice-based crypto

Words are hard

“So... How’s it going with your isonegies?”

— a lattice-based crypto researcher

...I mean, a carbon-based researcher who works on lattice-based crypto

Mnemonic:

“I so genius!”

Diffie–Hellman key exchange ’76

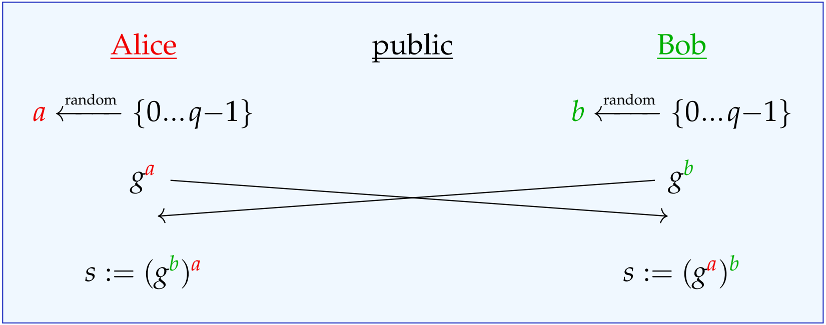
Public parameters:

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

Diffie–Hellman key exchange ’76

Public parameters:

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



Alice

public

Bob

$a \xleftarrow{\text{random}} \{0\ldots q-1\}$

$b \xleftarrow{\text{random}} \{0\ldots q-1\}$

g^a

g^b

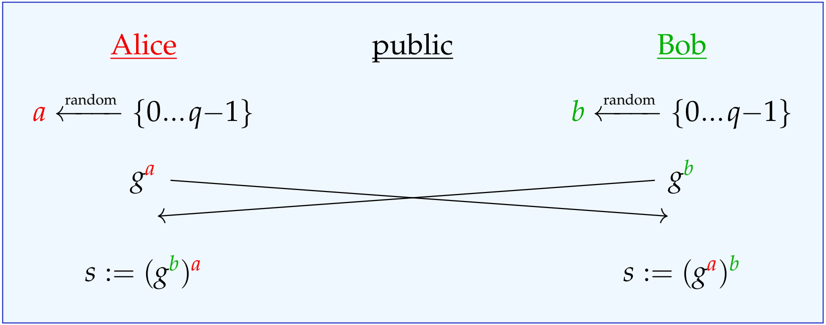
$s := (g^b)^a$

$s := (g^a)^b$

Diffie–Hellman key exchange '76

Public parameters:

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



Alice

public

Bob

$a \xleftarrow{\text{random}} \{0 \ldots q-1\}$

$b \xleftarrow{\text{random}} \{0 \ldots q-1\}$

g^a

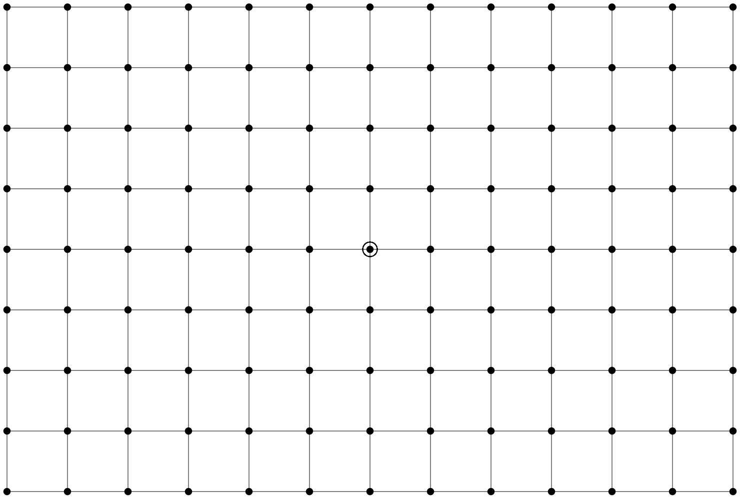
g^b

$s := (g^b)^a$

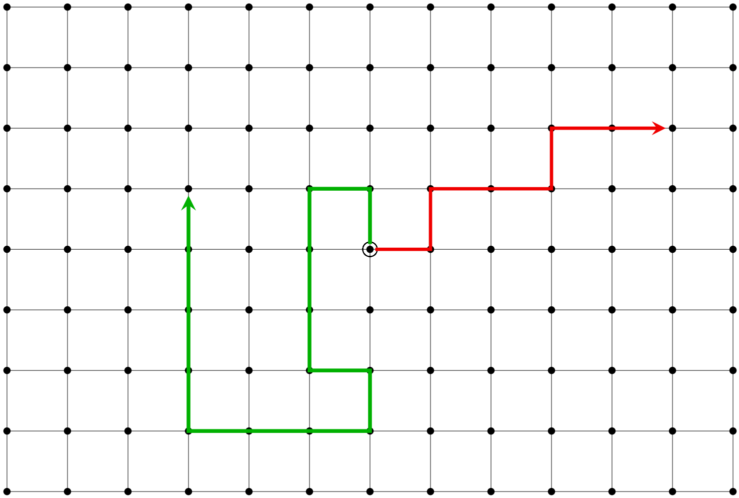
$s := (g^a)^b$

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

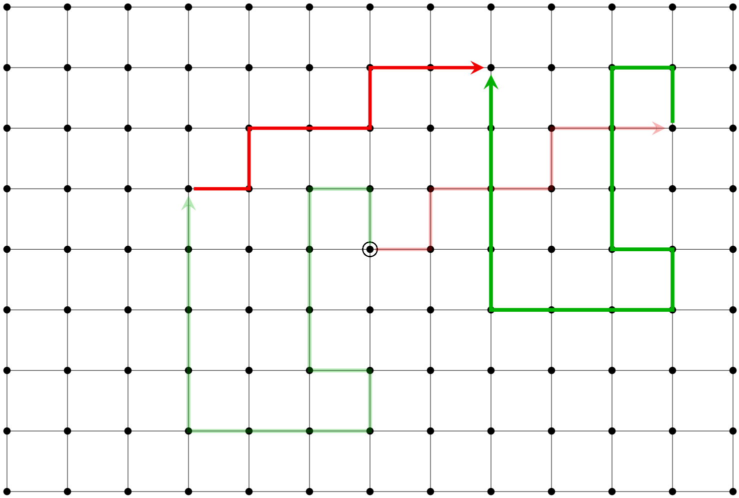
Graph walking Diffie–Hellman?



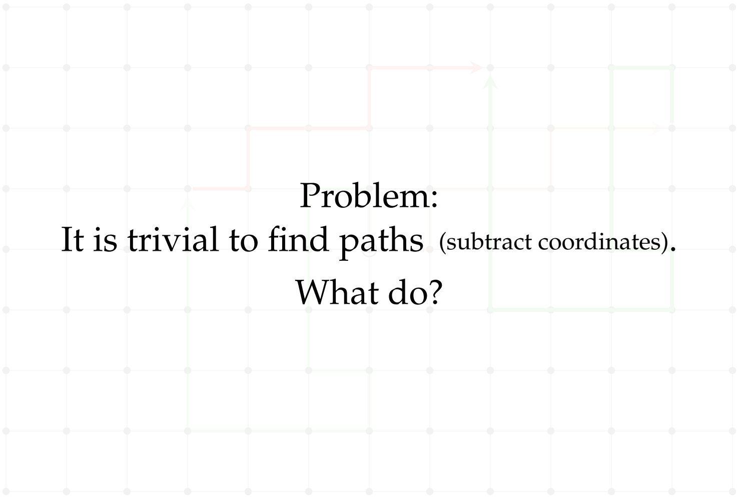
Graph walking Diffie–Hellman?



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 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 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 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!

There are several more-or-less equivalent viewpoints.
I will focus on one of them, hence omit many *fun* details.
Please ask me about stuff!

Stand back!

We're going to do math.

(worry not: only 4 ~~tough~~ exciting slides ahead!)

Math slide #1: Elliptic curves *(nodes)*

An elliptic curve (modulo details) is given by an equation

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

A point on E is a solution to this equation *or* the ‘fake’ point ∞.

Math slide #1: Elliptic curves *(nodes)*

An elliptic curve (modulo details) is given by an equation

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

A point on E is a solution to this equation *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!

Math slide #2: Isogenies *(edges)*

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

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

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

Math slide #2: Isogenies *(edges)*

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

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

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

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

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

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

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

Math slide #2: Isogenies *(edges)*

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

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

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

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

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

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

Math slide #2: Isogenies *(edges)*

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

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

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

Example #3: $(x, y) \mapsto \left(\frac{x^3-4x^2+30x-12}{(x-2)^2}, \frac{x^3-6x^2-14x+35}{(x-2)^3} \cdot y \right)$ defines a degree-3 isogeny of the elliptic curves

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

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

Math slide #3: Fields of definition

Until now: Everything over the algebraic closure.
For arithmetic, we need to know which fields objects live in.

Math slide #3: Fields of definition

Until now: Everything over the algebraic closure.
For arithmetic, we need to know which fields objects live in.

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

Math slide #3: Fields of definition

Until now: Everything over the algebraic closure.
For arithmetic, we need to know which fields objects live in.

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

For E defined over k, let $E(k)$ be the points of E defined over k.

Math slide #4: 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 #4: 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 <u>*supersingular*</u> 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 positive, pairwise coprime integers.

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

- isomorphism classes of supersingular elliptic curves
- with equivalence classes[1] of ℓ-isogenies ($\ell \in S$) as edges;

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

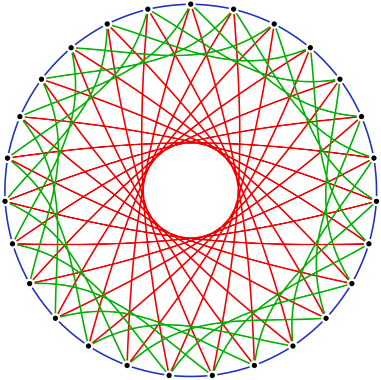
[1] Two isogenies $\varphi \colon E \to E'$ and $\psi \colon E \to E''$ are identified if $\psi = \iota \circ \varphi$ for some isomorphism $\iota \colon E' \to E''$.

The beauty and the beast

Components of the isogeny graphs look as follows:

The beauty and the beast

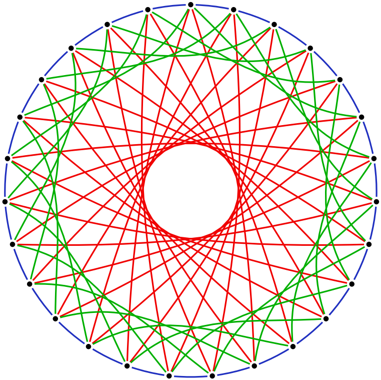
Components of the isogeny graphs look as follows:



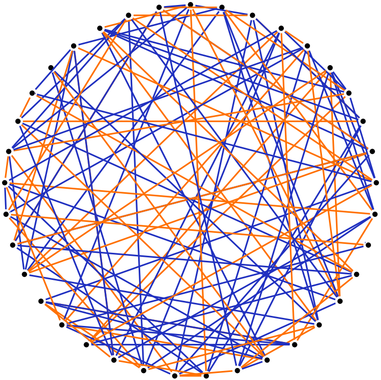
$S = \{3, 5, 7\}$, $q = 419$

The beauty and the beast

Components of the isogeny graphs look as follows:



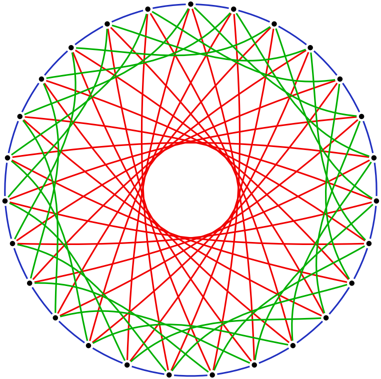
$S = \{3, 5, 7\}$, $q = 419$



$S = \{2, 3\}$, $q = 431^2$

The beauty and the beast

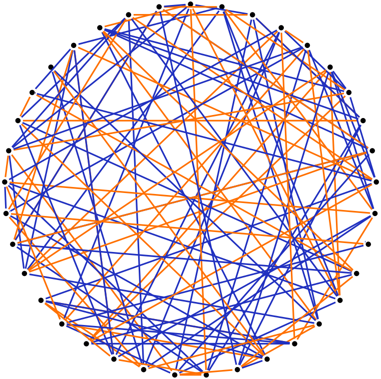
At this time, there are two distinct families of systems:



$q = p$

CSIDH [ˈsiːˌsaɪd]

`https://csidh.isogeny.org`



$q = p^2$

SIDH

`https://sike.org`

[ˈsiːˌsaɪd]

CSIDH

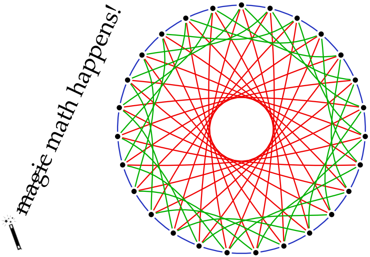
- Let $p = 4\prod_{i=1}^{n} \ell_i - 1$ be a prime; the ℓ_i distinct odd primes.

CSIDH

- Let $p = 4\prod_{i=1}^{n} \ell_i - 1$ be a prime; the ℓ_i distinct odd primes.
- Let $X = \{$supersingular $y^2 = x^3 + Ax^2 + x$ defined over $\mathbb{F}_p\}$.
- We consider the graph of $\{\ell_1, ..., \ell_n\}$-isogenies on X.

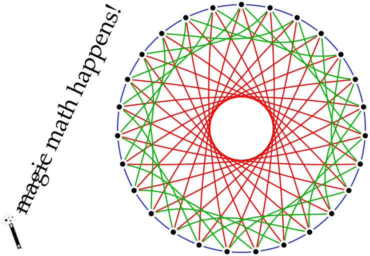
CSIDH

- Let $p = 4\prod_{i=1}^{n} \ell_i - 1$ be a prime; the ℓ_i distinct odd primes.
- Let $X = \{$supersingular $y^2 = x^3 + Ax^2 + x$ defined over $\mathbb{F}_p\}$.
- We consider the graph of $\{\ell_1, \ldots, \ell_n\}$-isogenies on X.



CSIDH

- Let $p = 4\prod_{i=1}^{n} \ell_i - 1$ be a prime; the ℓ_i distinct odd primes.
- Let $X = \{$supersingular $y^2 = x^3 + Ax^2 + x$ defined over $\mathbb{F}_p\}$.
- We consider the graph of $\{\ell_1, \ldots, \ell_n\}$-isogenies on X.

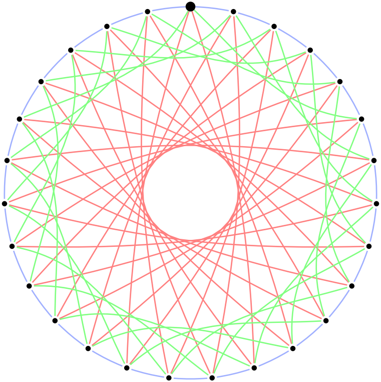


- Walking 'left' and 'right' on any ℓ_i-subgraph is efficient.

CSIDH key exchange

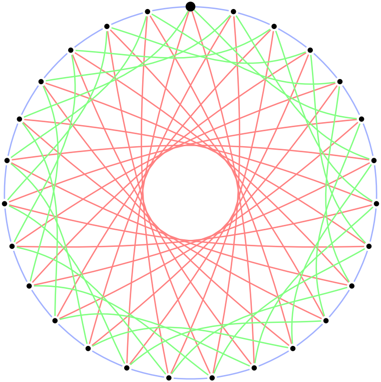
Alice

$[+, +, -, -]$



Bob

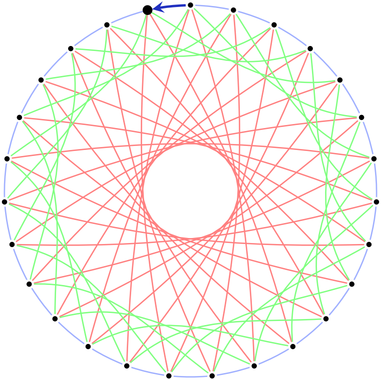
$[-, +, -, -]$



CSIDH key exchange

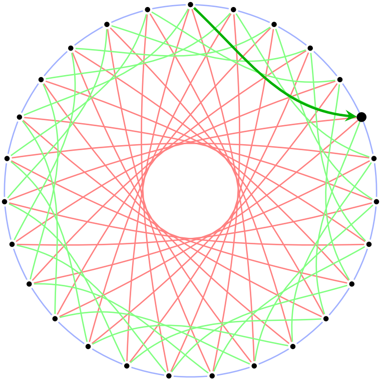
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



CSIDH key exchange

Alice

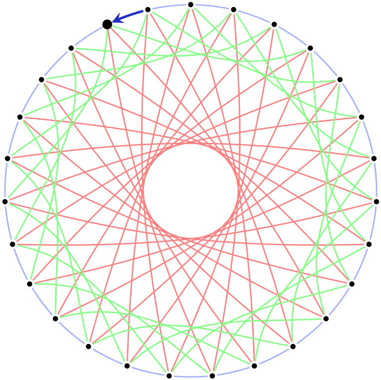
[+, +, −, −]

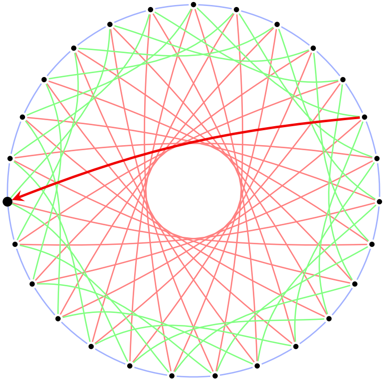
↑

Bob

[−, +, −, −]

↑

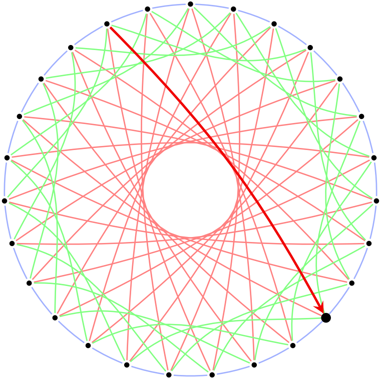




CSIDH key exchange

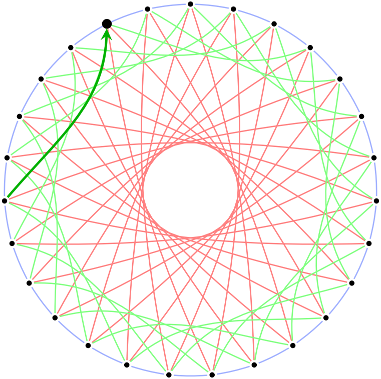
Alice

[+, +, −, −]
↑



Bob

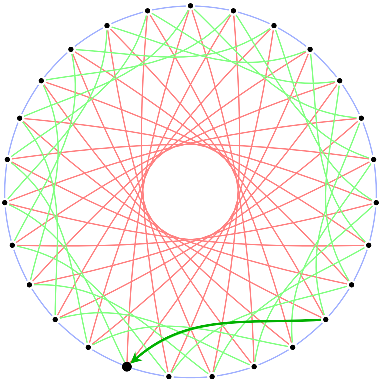
[−, +, −, −]
↑



CSIDH key exchange

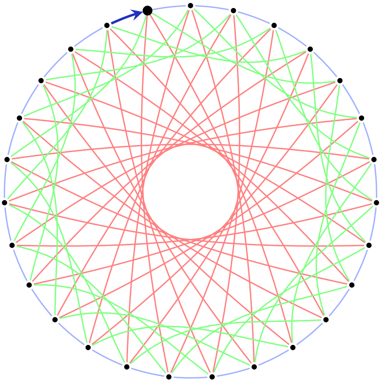
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



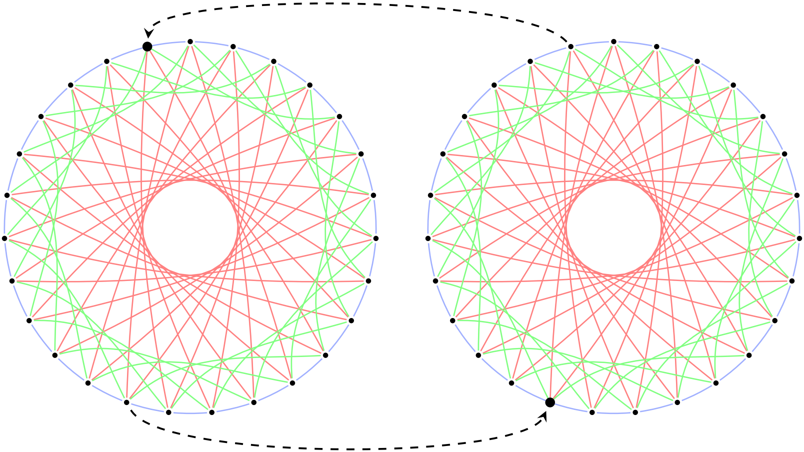
CSIDH key exchange

Alice

[+, +, −, −]

Bob

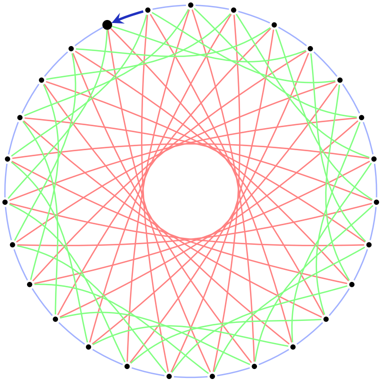
[−, +, −, −]



CSIDH key exchange

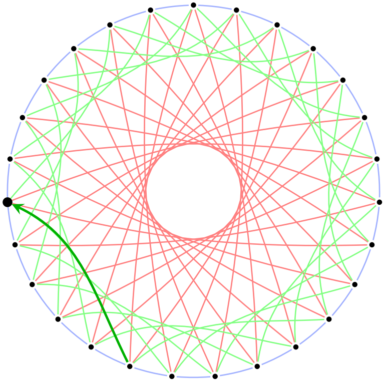
Alice

[+, +, −, −]
↑



Bob

[−, +, −, −]
↑



CSIDH key exchange

Alice

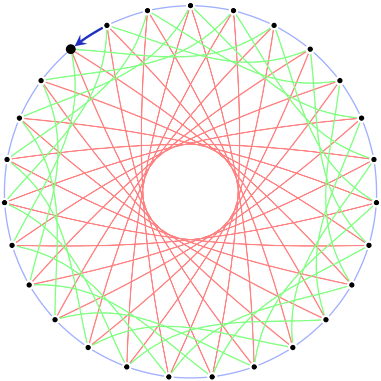
[+, +, −, −]

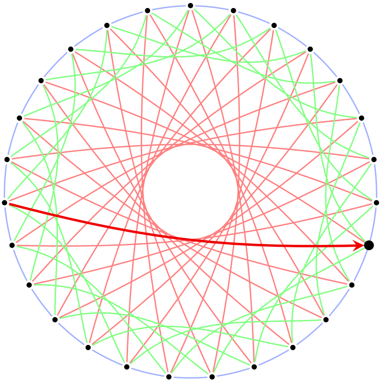
↑

Bob

[−, +, −, −]

↑



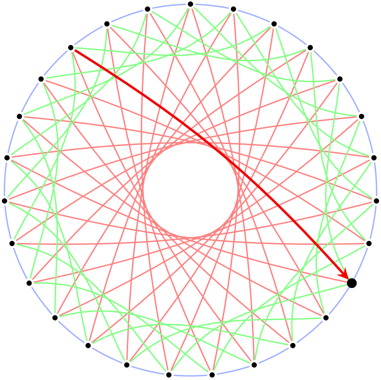


CSIDH key exchange

Alice

[+, +, −, −]

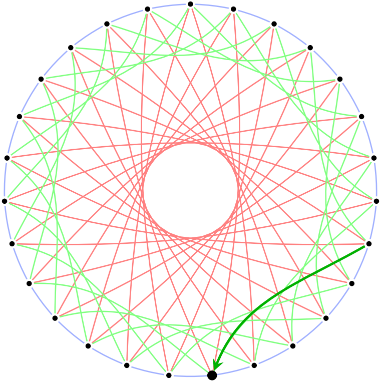
↑



Bob

[−, +, −, −]

↑



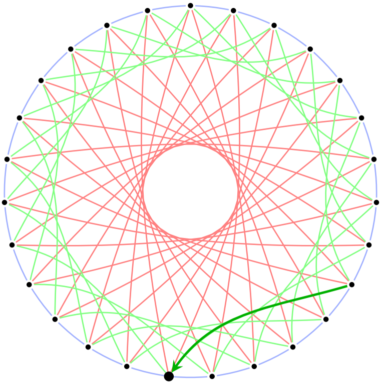
CSIDH key exchange

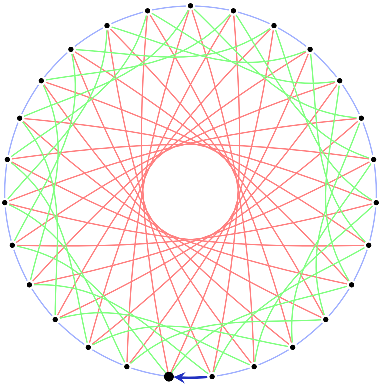
Alice

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

Bob

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

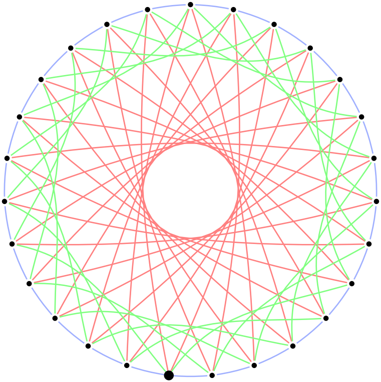




CSIDH key exchange

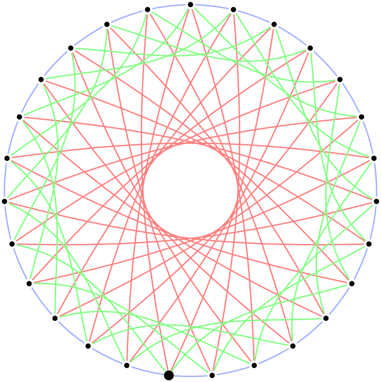
Alice

[+, +, −, −]



Bob

[−, +, −, −]



Has anyone seen my class group 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 $(+1,\quad 0,-3) \in \mathbb{Z}^3$.

Has anyone seen my class group 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!

Has anyone seen my class group 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!

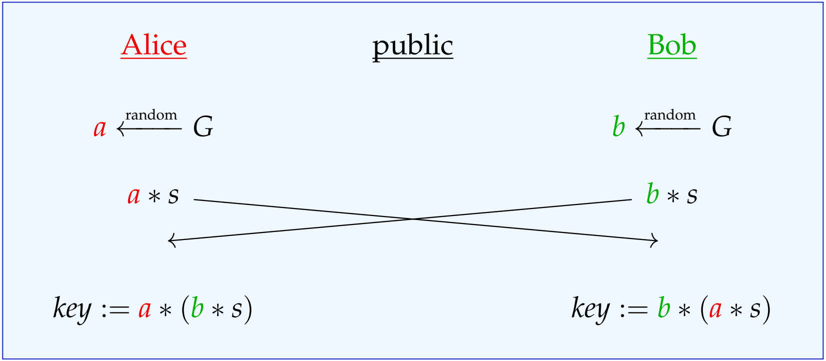
This action is transitive (for big enough n), but not free.
*Obviously**, quotienting out vectors which act trivially yields a group isomorphic to the ideal-class group $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$.

(This is because the curves in X have $\mathbb{F}_p$-endomorphism ring $\mathbb{Z}[\pi] \cong \mathbb{Z}[\sqrt{-p}]$. A prime ideal in $\mathbb{Z}[\pi]$ of norm ℓ corresponds to one of two eigenspaces of the Frobenius endomorphism π on the ℓ-torsion, which correspond to horizontal ℓ-isogenies that preserve the endomorphism ring.)

Cryptographic group actions

Previous slide: Free, transitive group action of $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ on X.

Like in the CSIDH example before, we *generally* get a DH-like key exchange from a group action $G \times S \to S$:



Alice

public

Bob

$a \xleftarrow{\text{random}} G$

$b \xleftarrow{\text{random}} G$

$a * s$

$b * s$

$key := a * (b * s)$

$key := b * (a * s)$

Why no Shor?

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

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

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

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a smooth-degree isogeny $E \to E'$.
Given $E, E' \in X$, find a smooth ideal $\mathfrak{a}$ of $\mathbb{Z}[\sqrt{-p}]$ with $[\mathfrak{a}]E = E'$.

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a smooth-degree isogeny $E \to E'$.
Given $E, E' \in X$, find a smooth ideal $\mathfrak{a}$ of $\mathbb{Z}[\sqrt{-p}]$ with $[\mathfrak{a}]E = E'$.

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

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

Security of CSIDH

Core problem:
Given $E, E' \in X$, find a smooth-degree isogeny $E \to E'$.
Given $E, E' \in X$, find a smooth ideal $\mathfrak{a}$ of $\mathbb{Z}[\sqrt{-p}]$ with $[\mathfrak{a}]E = E'$.

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

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

Solving abelian hidden shift breaks CSIDH.

$\rightsquigarrow$ quantum subexponential attack (Kuperberg's algorithm).

Can we avoid Kuperberg's algorithm?

With great commutative group action
comes great subexponential attack.

Can we avoid Kuperberg's algorithm?

With great commutative group action
comes great subexponential attack.

The supersingular isogeny graph over $\mathbb{F}_{p^2}$ has less structure.

- SIDH uses the full $\mathbb{F}_{p^2}$-isogeny graph. No group action!

Can we avoid Kuperberg's algorithm?

With great commutative group action
comes great subexponential attack.

The supersingular isogeny graph over $\mathbb{F}_{p^2}$ has less structure.

- SIDH uses the full $\mathbb{F}_{p^2}$-isogeny graph. No group action!

- Problem: also no more intrinsic sense of direction.

  "It all bloody looks the same!" — a famous isogeny cryptographer

$\rightsquigarrow$ need extra information to let Alice&Bob's walks commute.

Math slide #5: 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 called 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')

Math slide #5: 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 called 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')

Math slide #5: 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 called 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')

Now:
SIDH

(...whose name doesn't allow for nice pictures of beaches...)

Wikipedia about SIDH...

"While several steps of SIDH involve complex isogeny calculations, the overall flow of SIDH for parties A and B is straightforward for those familiar with a Diffie–Hellman key exchange or its elliptic curve variant. [...]

Wikipedia about SIDH...

“While several steps of SIDH involve complex isogeny calculations, the overall flow of SIDH for parties A and B is straightforward for those familiar with a Diffie–Hellman key exchange or its elliptic curve variant. [...]

Setup.

1. A prime of the form $p = w_A^{e_A} \cdot w_B^{e_B} \cdot f \pm 1$.
2. A supersingular elliptic curve E over $\mathbb{F}_{p^2}$.
3. Fixed elliptic points P_A, Q_A, P_B, Q_B on E.
4. The order of P_A and Q_A is $(w_A)^{e_A}$.
5. The order of P_B and Q_B is $(w_B)^{e_B}$.

Key exchange. [...]

1A. A generates two random integers $m_A, n_A < (w_A)^{e_A}$.
2A. A generates $R_A := m_A \cdot (P_A) + n_A \cdot (Q_A)$.
3A. A uses the point R_A to create an isogeny mapping $\phi_A : E \to E_A$ and curve E_A isogenous to E.
4A. A applies ϕ_A to P_B and Q_B to form two points on E_A : $\phi_A(P_B)$ and $\phi_A(Q_B)$.
5A. A sends to B E_A, $\phi_A(P_B)$, and $\phi_A(Q_B)$.
1B–4B. Same as A1 through A4, but with A and B subscripts swapped.
5B. B sends to A E_B, $\phi_B(P_A)$, and $\phi_B(Q_A)$.
6A. A has $m_A, n_A, \phi_B(P_A)$, and $\phi_B(Q_A)$ and forms $S_{BA} := m_A(\phi_B(P_A)) + n_A(\phi_B(Q_A))$.
7A. A uses S_{BA} to create an isogeny mapping ψ_{BA}.
8A. A uses ψ_{BA} to create an elliptic curve E_{BA} which is isogenous to E.
9A. A computes $K :=$ j-invariant (j_{BA}) of the curve E_{BA}.
6B. Similarly, B has $m_B, n_B, \phi_A(P_B)$, and $\phi_A(Q_B)$ and forms $S_{AB} = m_B(\phi_A(P_B)) + n_B(\phi_A(Q_B))$.
7B. B uses S_{AB} to create an isogeny mapping ψ_{AB}.
8B. B uses ψ_{AB} to create an elliptic curve E_{AB} which is isogenous to Ek
9B. B computes $K :=$ j-invariant (j_{AB}) of the curve E_{AB}.

The curves E_{AB} and E_{BA} are guaranteed to have the same j-invariant.”

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_{A}"] \arrow[d, "\varphi_{B}"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.
- Alice somehow obtains $A' := \varphi_B(A)$. (Similar for Bob.)

SIDH: High-level view

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.
- Alice somehow obtains $A' := \varphi_B(A)$. (Similar for Bob.)
- They both compute the shared secret

  $$(E/B)/A' \cong E/\langle A, B\rangle \cong (E/A)/B'.$$

SIDH's auxiliary points

Previous slide: "Alice somehow obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B. Hm.

SIDH's auxiliary points

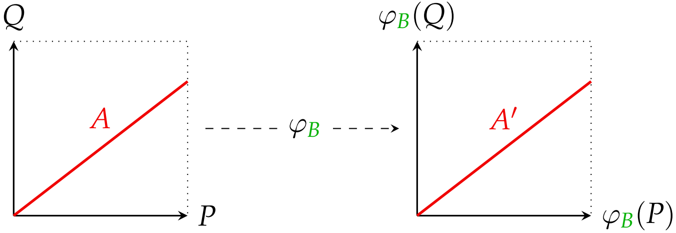
Previous slide: "Alice somehow obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B. Hm.

Solution: φ_B is a group homomorphism!

- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.
- Bob includes $\varphi_B(P)$ and $\varphi_B(Q)$ in his public key.

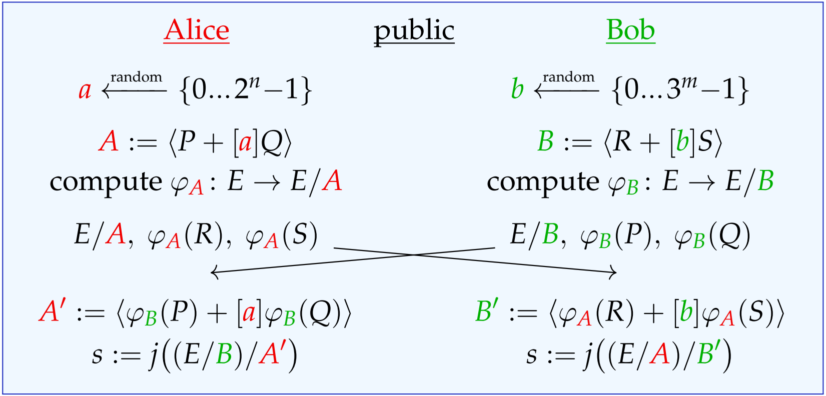
$\Longrightarrow$ Now Alice can compute A' as $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$!



SIDH in one slide

Public parameters:

- a large prime $p = 2^n 3^m - 1$ and a supersingular $E/\mathbb{F}_p$
- bases (P, Q) and (R, S) of $E[2^n]$ and $E[3^m]$



Alice

public

Bob

$a \xleftarrow{\text{random}} \{0 \ldots 2^n - 1\}$

$b \xleftarrow{\text{random}} \{0 \ldots 3^m - 1\}$

$A := \langle P + [a]Q \rangle$

compute $\varphi_A : E \to E/A$

$B := \langle R + [b]S \rangle$

compute $\varphi_B : E \to E/B$

$E/A,\ \varphi_A(R),\ \varphi_A(S)$

$E/B,\ \varphi_B(P),\ \varphi_B(Q)$

$A' := \langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$

$s := j\big((E/B)/A'\big)$

$B' := \langle \varphi_A(R) + [b]\varphi_A(S) \rangle$

$s := j\big((E/A)/B'\big)$

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
(Alice & Bob can choose from about $\sqrt{p}$ secret keys each.)

[1] `https://ia.cr/2019/103`

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
(Alice & Bob can choose from about $\sqrt{p}$ secret keys each.)

Classical attacks:

- Cannot reuse keys without extra caution.
- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/4})$ time & space.
- Collision finding: $\tilde{\mathcal{O}}(p^{3/8}/\sqrt{memory}/cores)$.

[1] `https://ia.cr/2019/103`

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
(Alice & Bob can choose from about $\sqrt{p}$ secret keys each.)

Classical attacks:

- Cannot reuse keys without extra caution.
- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/4})$ time & space.
- Collision finding: $\tilde{\mathcal{O}}(p^{3/8}/\sqrt{memory}/cores)$.

Quantum attacks:

- Claw finding: claimed $\tilde{\mathcal{O}}(p^{1/6})$. New paper[1] says $\tilde{\mathcal{O}}(p^{1/4})$:

  "An adversary with enough quantum memory to run Tani's algorithm with the query-optimal parameters could break SIKE faster by using the classical control hardware to run van Oorschot–Wiener."

[1] https://ia.cr/2019/103

Open and half-open questions

CSIDH:

How costly is breaking CSIDH with Kuperberg's algorithm?

Is Kuperberg's algorithm optimal for abelian hidden shift?

Are there any non-generic quantum attacks?

SIDH:

Do the points $\varphi_B(P), \varphi_B(Q)$ reveal too much information?

Can we phrase SIDH as a hidden-subgroup problem?

Are there any non-generic quantum attacks?

Thank you!