Constructing broken SIDH parameters: a tale of De Feo, Jao, and Plût's serendipity

Chloe Martindale

University of Bristol

University of Waterloo, 6 November 2020

Joint work with Péter Kutas, Lorenz Panny, Christophe Petit, Victoria de Quehen, and Kate Stange

What is this all about?

Diffie–Hellman key exchange '76

Public parameters:

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

Diffie–Hellman key exchange ’76

Public parameters:

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

<u>Alice</u>	<u>public</u>	<u>Bob</u>
$a \xleftarrow{\text{random}} \{0 \ldots p-1\}$		$b \xleftarrow{\text{random}} \{0 \ldots p-1\}$
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 also elliptic curves)
- an element $g \in G$ of prime order p

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

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

Diffie–Hellman key exchange '76

Public parameters:

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

BROKEN!

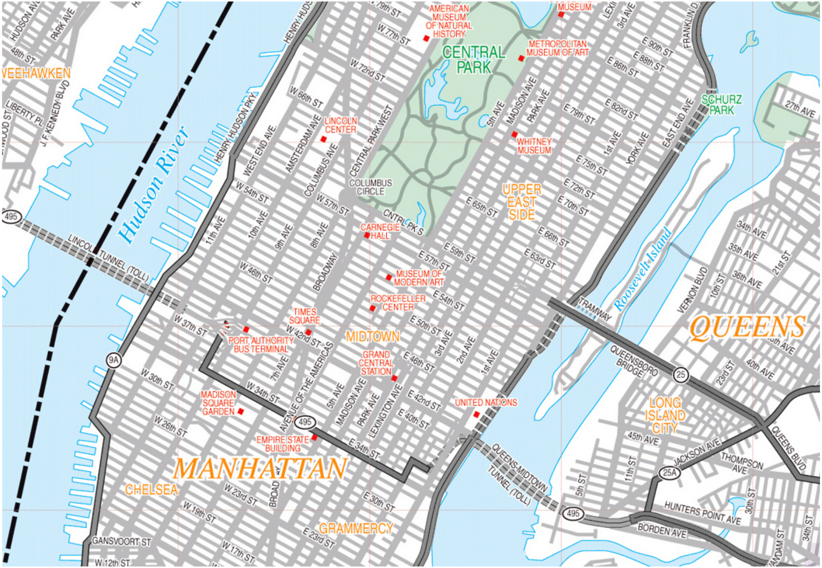
Alice	Bob
$a \xleftarrow{\text{random}} \{0$	$b \xleftarrow{\text{random}} \{0 \dots p-1\}$
	g^b
	$s := (g^a)^b$

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

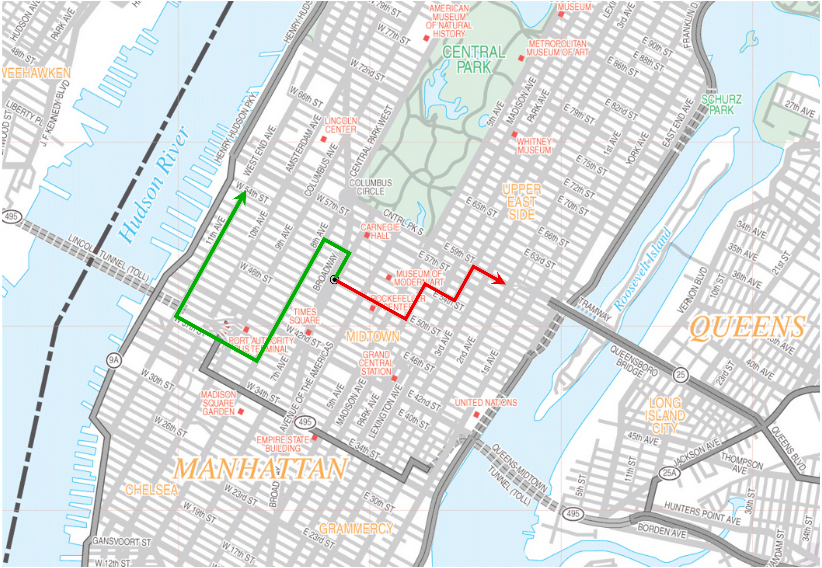
Quantum cryptoapocalypse

- Diffie-Hellman relies on the Discrete Logarithm Problem being hard.
  - Read: taking (discrete) logarithms should be much slower than exponentiating.
- Shor's quantum algorithm solves the discrete logarithm problem in polynomial time.
  - Read: with access to a quantum computer, taking discrete logarithms is about as fast as exponentiation.
- Quantum computers that are sufficiently large and stable do not yet exist (probably).
- But they are likely to be only a few years away...

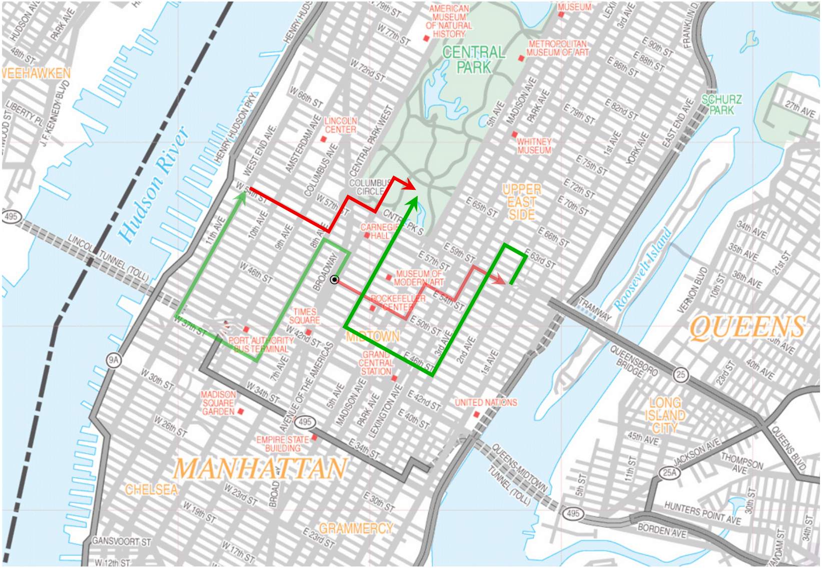
Graph walking Diffie–Hellman?



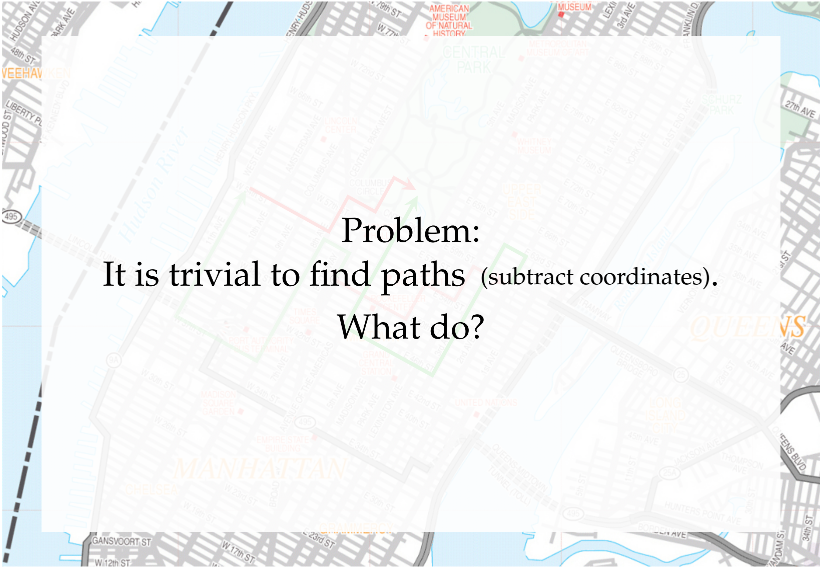
Graph walking Diffie–Hellman?



Graph walking Diffie–Hellman?



Graph walking Diffie–Hellman?



Problem:

It is trivial to find paths (subtract coordinates).

What do?

Big picture

- <u>Isogenies</u> 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: 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 ∞.

Maths background #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 easy to compute.

Maths background #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!

Maths background #2: 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 #2: 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.

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

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

Maths background #2: 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.

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 $\{\infty\}$.

Maths background #2: 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.

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 $\{(2, 9), (2, -9), \infty\}$.

Maths background #2: 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 #2: 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 #3: Fields of definition

Until now: Everything over the algebraic closure.

For arithmetic, we need to know which fields objects live in.

Maths background #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 of its equation/formula lie in k.

Maths background #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 of its equation/formula lie in k.

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

Maths background #4: 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 #4: 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 #4: 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')

Maths background #5: Supersingular isogeny graphs

Let p be a prime and q a power of p.

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!

Maths background #5: Supersingular isogeny graphs

Let p be a prime and q a power of p.

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!

Our supersingular isogeny graph over $\mathbb{F}_{p^2}$ will consist of:

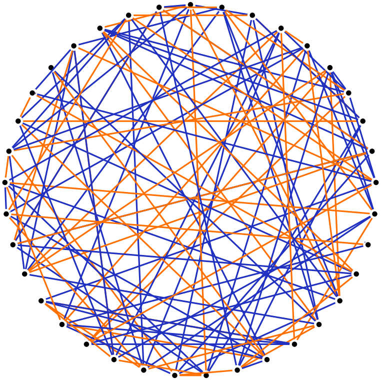
- vertices given by supersingular elliptic curves (up to isomorphism),
- edges given by equivalence classes[1] of 2 and 3-isogenies,

both defined over $\mathbb{F}_{p^2}$.

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

Graph-walking Diffie-Hellman?

The isogeny graph looks like this:



$p = 431$

Now:
SIDH

Supersingular Isogeny Diffie–Hellman

De Feo, Jao, Plût 2011

Diffie-Hellman: High-level view

$$
\begin{tikzcd}
g \arrow[r] \arrow[d] & g^{a} \arrow[d] \\
g^{b} \arrow[r] & g^{a*b}
\end{tikzcd}
$$

SIDH: High-level view

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

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 $\varphi_B \colon E \to E/B$.

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 $\varphi_A : E \to E/A$; Bob computes $\varphi_B : E \to E/B$.
- 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 $\varphi_B \colon E \to E/B$.
- 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 $\varphi_B \colon E \to E/B$.
- 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 $A' := \varphi_B(A)$."

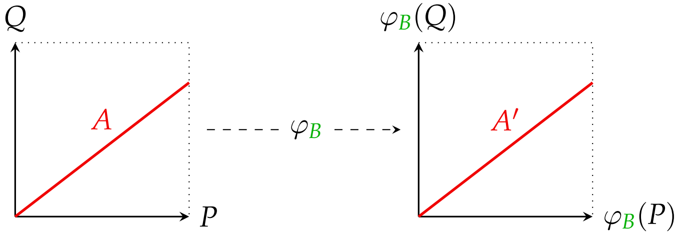
Alice knows only A, Bob knows only φ_B. Hm.

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!

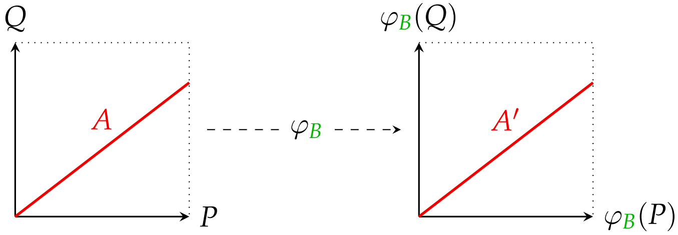


SIDH's auxiliary points

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_A, Q_A) and (P_B, Q_B) of $E[2^n]$ and $E[3^m]$

<u>Alice</u>	<u>public</u>	<u>Bob</u>
$a \xleftarrow{\text{random}} \{0 \ldots 2^n - 1\}$		$b \xleftarrow{\text{random}} \{0 \ldots 3^m - 1\}$
$A := \langle P_A + [a]Q_A \rangle$		$B := \langle P_B + [b]Q_B \rangle$
compute $\varphi_A : E \to E/A$		compute $\varphi_B : E \to E/B$
$E/A,\ \varphi_A(P_B),\ \varphi_A(Q_B)$		$E/B,\ \varphi_B(P_A),\ \varphi_B(Q_A)$
$A' := \langle \varphi_B(P_A) + [a]\varphi_B(Q_A) \rangle$		$B' := \langle \varphi_A(P_B) + [b]\varphi_A(Q_B) \rangle$
$s := j\big((E/B)/A'\big)$		$s := j\big((E/A)/B'\big)$

Attack by: given public info, find secret key–φ_A or just A.

Torsion-point attacks on SIDH

'Breaking SIDH' means:

Given

- supersingular public elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ connected by a secret 2^n-degree isogeny $\varphi_A : E_0 \to E_A$, and
- the action of φ_A on the 3^m-torsion of E_0,

finding the secret key recover φ_A.

Torsion-point attacks on SIDH

'Breaking SIDH' means:

Given

- supersingular public elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ connected by a secret 2^n-degree isogeny $\varphi_A : E_0 \to E_A$, and
- the action of φ_A on the 3^m-torsion of E_0,

finding the secret key recover φ_A.

2016 Galbraith, Petit, Shani, Ti: knowledge of $\mathrm{End}(E_0)$ and $\mathrm{End}(E_A)$ is sufficient to efficiently break it.

2017 Petit: If $E_0 : y^2 = x^3 + x$ and $3^m > 2^{4n} > p^4$, then we can construct non-scalar $\theta \in \mathrm{End}(E_A)$ and efficiently break it.

But in SIDH, $3^m \approx 2^n \approx \sqrt{p}$.

Torsion-point attacks on SIDH

'Breaking SIDH' means:

Given

- supersingular public elliptic curves $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ connected by a secret D-degree isogeny $\varphi_A : E_0 \to E_A$, and
- the action of φ_A on the T-torsion of E_0,

finding the secret key recover φ_A.

2016 Galbraith, Petit, Shani, Ti: knowledge of $\mathrm{End}(E_0)$ and $\mathrm{End}(E_A)$ is sufficient to efficiently break it.

2017 Petit: If $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$, then we can construct non-scalar $\theta \in \mathrm{End}(E_A)$ and efficiently break it.

But in SIDH, $T \approx D \approx \sqrt{p}$.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd}
E_0 \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"]
\end{tikzcd}$$

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[r, bend left, "\varphi_A"] & E \arrow[l, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"]
\end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"]
\end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A} \ (+[n])"] \end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$
\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"]
\end{tikzcd}
$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.
- If there exist ι, n such that $\deg(\theta) = T$, then can completely determine θ, and φ_A.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[r, bend left, "\varphi_A"] & E \arrow[l, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.
- If there exist ι, n, ϵ such that $\deg(\theta) = \epsilon T$, then can completely determine θ, and φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^4 > p^4$:
finding the secret isogeny φ_A of degree D.

$$
\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[r, bend left, "\varphi_A"] & E \arrow[l, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"]
\end{tikzcd}
$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.
- If there exist ι, n, ϵ such that $\deg(\theta) = \epsilon T$, then can completely determine θ, and φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.

- We can heuristically do this for polynomially small ϵ when $T > D^4 > p^4$. *

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^2 > p^2$:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.
- If there exist ι, n, ϵ such that $\deg(\theta) = \epsilon T^2$, then can completely determine θ, and φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.

- We can heuristically do this for polynomially small ϵ when $T > D^2 > p^2$.*

* See also `https://eprint.iacr.org/2019/1333`.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$ and $T > D^2 > p^2$:
finding the secret isogeny φ_A of degree D.

$$
\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] && E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"]
\end{tikzcd}
$$

- We can choose $\iota \in \mathrm{End}(E_0)$ (for simplicity: of trace zero).
- Know the action of φ_A (and $\widehat{\varphi_A}$) on the T-torsion.
- Know: $\deg(\theta) = D^2 \deg(\iota) + n^2$.
- If there exist ι, n, ϵ such that $\deg(\theta) = \epsilon T^2$, then can completely determine θ, and φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.

- We can heuristically do this for polynomially small ϵ when $T > D^2 > p^2$.*

---

* See also `https://eprint.iacr.org/2019/1333`.

Improvements on torsion-point attacks

Know:

- $\epsilon T^2 = \deg(\theta) = D^2 \deg(\iota) + n^2.$

Improvements on torsion-point attacks

Know:

- $\epsilon T^2 = \deg(\theta) = D^2 \deg(\iota) + n^2$.
- $\iota \in \mathrm{End}(E_0)$ and $E_0 : y^2 = x^3 + x \rightsquigarrow \deg(\iota) = pa^2 + pb^2 + c^2$

  (modulo details)

Improvements on torsion-point attacks

Know:

- $\epsilon T^2 = \deg(\theta) = D^2 \deg(\iota) + n^2$.
- $\iota \in \mathrm{End}(E_0)$ and $E_0 : y^2 = x^3 + x \rightsquigarrow \deg(\iota) = pa^2 + pb^2 + c^2$ (modulo details)

Algorithm is in 2 parts:

1. Find $a, b, c, n, \epsilon \in \mathbb{Z}$ with ϵ small such that $D^2(pa^2 + pb^2 + c^2) + n^2 = \epsilon T^2$.

Improvements on torsion-point attacks

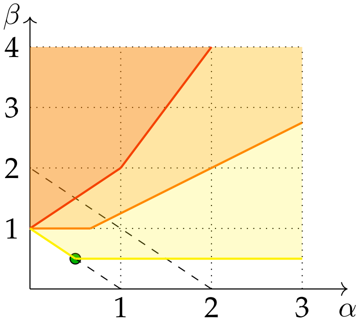
Know:

- $\epsilon T^2 = \deg(\theta) = D^2 \deg(\iota) + n^2$.
- $\iota \in \mathrm{End}(E_0)$ and $E_0 : y^2 = x^3 + x \rightsquigarrow \deg(\iota) = pa^2 + pb^2 + c^2$ (modulo details)

Algorithm is in 2 parts:

1. Find $a, b, c, n, \epsilon \in \mathbb{Z}$ with ϵ small such that $D^2(pa^2 + pb^2 + c^2) + n^2 = \epsilon T^2$.
2. Reconstruct $\iota \in \mathrm{End}(E_0)$ with degree $pa^2 + pb^2 + c^2$ and use that to compute φ_A.

Improvements on torsion-point attacks



- $D \approx p^\alpha$, $T \approx p^\beta$.
- Below 1-1 dotted line: attacks SIDH group key exchange.
- Below 2-2 dotted line: attacks B-SIDH.[1]
- Polynomial-time attack, improved classical attack, improvemed quantum attack, SIDH.
- Left: our results. Right: your results, if...

[1] https://eprint.iacr.org/2019/1145.pdf

The equation of death

Open question:

For $\sqrt{p} \approx D \approx T$, and p large,
find $a, b, c, n, \epsilon \in \mathbb{Z}$ with $\epsilon \approx \sqrt{D^3 p}/T$ such that

$$D^2(pa^2 + pb^2 + c^2) + n^2 = \epsilon T^2$$

in time polynomial in $\log(p)$.

From torsion points to endomorphisms

The case of $E_0 : y^2 = x^3 + x$
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[r, bend left, "\varphi_A"] & E \arrow[l, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- Find φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.
- We can heuristically do this for polynomially small ϵ when $T > D^2 > p^2$.
- For $T \approx D \approx \sqrt{p}$, like in SIDH, $\epsilon \geq \sqrt{D^3 p}/T$.

From torsion points to endomorphisms

The case of specially constructed E_0:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- Find φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.
- We can heuristically do this for polynomially small ϵ when $T > D^2 > p^2$.
- For $T \approx D \approx \sqrt{p}$, like in SIDH, $\epsilon \geq \sqrt{D^3 p}/T$.

From torsion points to endomorphisms

The case of specially constructed E_0:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- Find φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.
- We can heuristically do this for polynomially small ϵ when $T > D^2$.
- For $T \approx D \approx \sqrt{p}$, like in SIDH, $\epsilon \geq \sqrt{D^3 p}/T$.

From torsion points to endomorphisms

The case of specially constructed E_0:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A} \ (+[n])"]
\end{tikzcd}$$

- Find φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.
- We can heuristically do this for polynomially small ϵ when $T > D^2$.
- For $T \approx D \approx \sqrt{p}$, like in SIDH, we can do this in time $p^{1/8}$.

From torsion points to endomorphisms

The case of specially constructed E_0:
finding the secret isogeny φ_A of degree D.

$$\begin{tikzcd} E_0 \arrow[loop left, "\iota"] \arrow[rr, bend left, "\varphi_A"] & & E \arrow[ll, bend left, "\widehat{\varphi_A}"] \arrow[loop right, "\theta = \varphi_A \circ \iota \circ \widehat{\varphi_A}\ (+[n])"] \end{tikzcd}$$

- Find φ_A, in time $O(\sqrt{\epsilon} \cdot \mathrm{polylog}(p))$.
- We can heuristically do this for polynomially small ϵ when $T > D^2$.
- For $T \approx D \approx \sqrt{p}$, like in SIDH, we can do this in time $p^{1/8}$.
- This is a square-root improvement over the previous best known attack.

SIDH is not broken

- Allowing for attack complexities up to the state-of-the-art, the balance of SIDH is exactly at the point where torsion-point attacks give no improvement.
- There are many specially constructed starting curves allowing for an attack, but probably none help with attacking SIDH proper.
- One more thing: you can also construct special base field primes to get efficient torsion point attacks
  (. . .which also don't apply to SIDH proper).

Thank you!

https://arxiv.org/abs/2005.14681