Making and breaking post-quantum cryptography from elliptic curves

Chloe Martindale

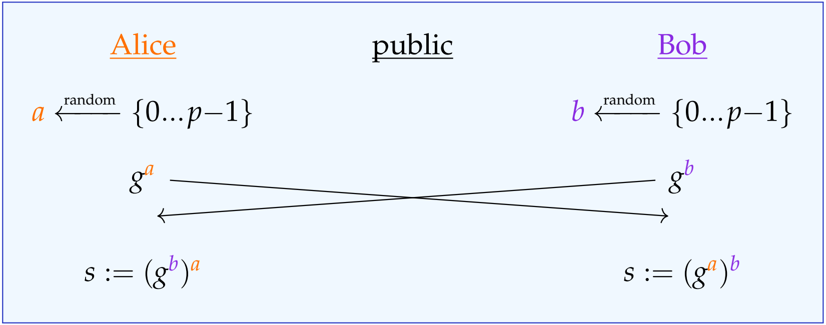
University of Bristol

11th June 2024

Recall: Diffie–Hellman key exchange '76

Public parameters:

- a finite group G (typically $\mathbb{F}_q^*$ or $E(\mathbb{F}_q)$)
- an element $g \in G$ of (large) 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

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

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

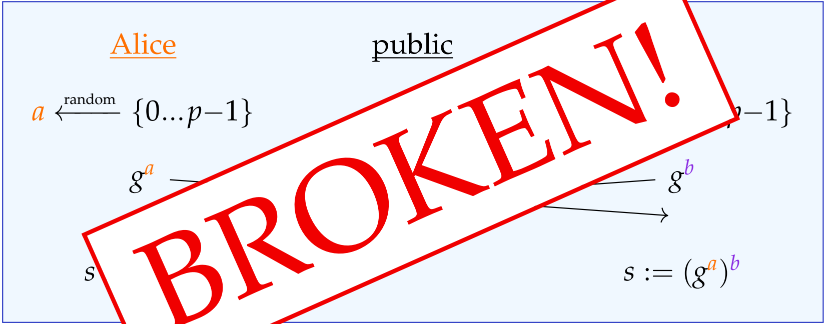
The Discrete Logarithm Problem, finding a given g and g^a, should be hard[1] in $\langle g \rangle$.

[1] Complexity (at least) subexponential in $\log(p)$.

Recall: Diffie–Hellman key exchange ‘76

Public parameters:

- a finite group G (typically $\mathbb{F}_q^*$ or $E(\mathbb{F}_q)$)
- an element $g \in G$ of (large) prime order p



BROKEN!

The Discrete Logarithm Problem, finding a given g and g^a, should be hard[1] in $\langle g \rangle$.

[1] Complexity (at least) subexponential in $\log(p)$.

Quantumifying Exponentiation

- Couveignes '97, Rostovtsev, Stolbunov '04: Idea to replace the Discrete Logarithm Problem: replace exponentiation

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

by a group action on a set.

Quantumifying Exponentiation

- Couveignes ‘97, Rostovtsev, Stolbunov ‘04: Idea to replace the Discrete Logarithm Problem: replace exponentiation

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

  by a group action on a set.

- Replace G by a set S of elliptic curves $/\mathbb{F}_q$ with commutative endomorphism ring $\mathcal{O}$.

Quantumifying Exponentiation

- Couveignes '97, Rostovtsev, Stolbunov '04: Idea to replace the Discrete Logarithm Problem: replace exponentiation

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

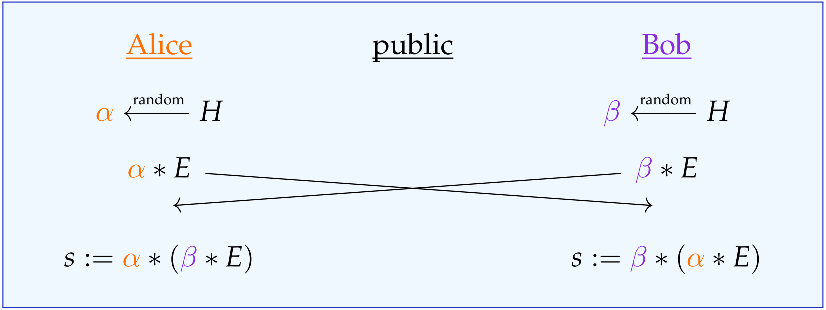
  by a group action on a set.
- Replace G by a set S of elliptic curves $/\mathbb{F}_q$ with commutative endomorphism ring $\mathcal{O}$.
- Replace $\mathbb{Z}$ by $\mathrm{Cl}(\mathcal{O})$; this will act freely and transitively on S via isogenies:

$$\begin{array}{ccc} \mathrm{Cl}(\mathcal{O}) \times S & \to & S \\ ([\mathbf{a}], E) & \mapsto & [\mathbf{a}] * E := E/\mathbf{a} \end{array}$$

Couveignes-Rostovstev-Stolbunov key exchange

Public parameters:

- the finite set S,
- an elliptic curve $E/\mathbb{F}_q \in S$,
- $H = \mathrm{Cl}(\mathrm{End}(E))$, that acts freely and transitively on S via $*$.



Alice

public

Bob

$\alpha \xleftarrow{\text{random}} H$

$\beta \xleftarrow{\text{random}} H$

$\alpha * E$

$\beta * E$

$s := \alpha * (\beta * E)$

$s := \beta * (\alpha * E)$

Finding α given E and $\alpha * E$, should be hard.[2]

[2] Complexity (at least) subexponential in $\log(\#S)$.

From CRS to CSIDH

1997 Couveignes proposes the now-CRS scheme.

- $S =$ ordinary elliptic curves/$\mathbb{F}_p$ with same end ring.
- Paper rejected and forgotten.

2004 Rostovstev, Stolbunov rediscover now-CRS scheme.

- Best known quantum and classical attacks are exponential.

2005 Kuperberg: quantum subexponential attack for the dihedral hidden subgroup problem.

2010 Childs, Jao, Soukharev apply Kuperberg to CRS.

- Secure parameters $\rightsquigarrow$ key exchange of 20 minutes.

2011 Jao, De Feo propose SIDH [more to come!].

2017 De Feo, Kieffer, Smith use modular curves to do a CRS key exchange in 8 minutes.

2018 Castryck, Lange, M., Panny, Renes propose CSIDH.

- CRS but with supersingular elliptic curves $/\mathbb{F}_p$.
- p constructed to make scheme efficient.
- Key exchange runs in 60ms.*

[ˈsiːˌsaɪd]

Evolution of key exchange

Diffie-Hellman

$$\begin{tikzcd}
& g_a = g^a \\
g \arrow[ur, "(-)^a"] \arrow[dr, "(-)^b"'] & \\
& g_b = g^b
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

Diffie-Hellman

$$\begin{tikzcd}
 & g_a = g^a \arrow[dd, dashed, shift right] \\
g \arrow[ur, "(-)^a"] \arrow[dr, "(-)^b"'] & \boxed{g_a, g_b} \\
 & g_b = g^b \arrow[uu, dashed, shift right]
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

Diffie-Hellman

$$\begin{tikzcd}
 & g_a = g^a \arrow[rd, "(-)^b"] \arrow[dddd, dashed, shift right=3] & \\
 & & (g_a)^b \arrow[d, equal] \\
g \arrow[ruu, "(-)^a"] \arrow[rdd, "(-)^b"'] & \boxed{g_a, g_b} & (g_b)^a \\
 & & \\
 & g_b = g^b \arrow[ruu, "(-)^a"'] \arrow[uuuu, dashed, shift right=3] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

CRS or CSIDH

$$\begin{tikzcd}
 & E_A = \alpha * E_0 \arrow[dddd, dashed, shift right] \arrow[dr, "\beta * (-)"] & \\
 & & \beta * E_A \\
E_0 \arrow[uur, "\alpha * (-)"] \arrow[ddr, "\beta * (-)"'] & \boxed{E_A, E_B} & \wr\| \\
 & & \alpha * E_B \\
 & E_B = \beta * E_0 \arrow[uuuu, dashed, shift right] \arrow[ur, "\alpha * (-)"'] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[ddrr, "\beta(-)"] \arrow[dddd, dashed, shift right=4] & & \\
 & & & \\
E_0 \arrow[uur, "\alpha(-)"] \arrow[ddr, "\beta(-)"'] & \boxed{E_A, E_B} & & E_{AB} \arrow[d, phantom, "\cong" sloped] \\
 & & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[urr, "\alpha(-)"'] \arrow[uuuu, dashed, shift right=4] & &
\end{tikzcd}$$

Colour code: Public, Alice’s secret, Bob’s secret

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[drr] \arrow[dddd, dashed, shift right] & & \\
 & & & E_{AB} \arrow[dd, phantom, "\cong" description, sloped] \\
E_0 \arrow[uur, "\ker\alpha = \langle A\rangle"] \arrow[ddr, "\ker\beta = \langle B\rangle"'] & \boxed{E_A, E_B} & & \\
 & & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[urr] \arrow[uuuu, dashed, shift right] & &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[dr, "\ker = \langle \alpha(B) \rangle"] \arrow[dddd, dashed, shift right=3] & \\
 & & E_{AB} \arrow[dd, phantom, "\wr\|"] \\
E_0 \arrow[uur, "\ker \alpha = \langle A \rangle"] \arrow[ddr, "\ker \beta = \langle B \rangle"'] & \boxed{E_A, E_B} & \\
 & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[ur, "\ker = \langle \beta(A) \rangle"'] \arrow[uuuu, dashed, shift right=3] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[dddd, dashed, shift right=3] \arrow[dr, "\ker = \langle \alpha(B) \rangle"] & \\
 & & E_{AB} \\
E_0 \arrow[uur, "\ker \alpha = \langle A \rangle"] \arrow[ddr, "\ker \beta = \langle B \rangle"'] & \boxed{E_A, E_B} & \wr\| \\
 & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[uuuu, dashed, shift right=3] \arrow[ur, "\ker = \langle \beta(A) \rangle"'] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret, ?!

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[dddd, dashed, shift right=3] \arrow[rd, "\ker = \langle \alpha(B) \rangle"] & \\
 & & E_{AB} \\
E_0 \arrow[ruu, "\ker \alpha = \langle A = P_A + aQ_A \rangle"] \arrow[rdd, "\ker \beta = \langle B \rangle"'] & \boxed{E_A, E_B} & \wr\| \\
 & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[uuuu, dashed, shift right=3] \arrow[ru, "\ker = \langle \beta(A) \rangle"'] &
\end{tikzcd}$$

Colour code: Public, Alice’s secret, Bob’s secret, ?!

Evolution of key exchange

From CRS to SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[dd, dashed, shift right] \arrow[dr, "\ker = \langle \alpha(B) \rangle"] & \\
 & & E_{AB} \arrow[dd, phantom, "\wr\|"] \\
P_A, Q_A, E_0, P_B, Q_B \arrow[uur, "\ker \alpha = \langle A = P_A + aQ_A \rangle"] \arrow[ddr, "\ker \beta = \langle B = P_B + bQ_B \rangle"'] & \boxed{E_A, E_B} & \\
 & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[uuuu, dashed, shift right] \arrow[ur, "\ker = \langle \beta(A) \rangle"'] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret, ?!

Evolution of key exchange

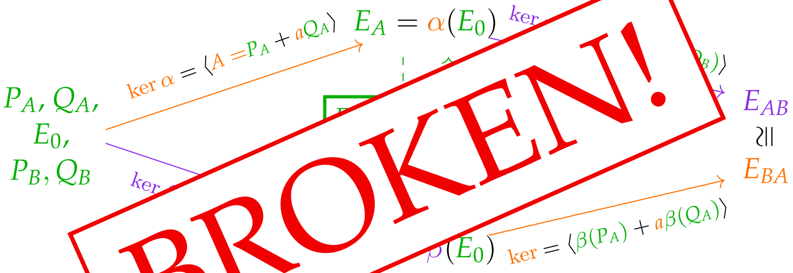
SIDH

$$\begin{tikzcd}
 & E_A = \alpha(E_0) \arrow[dd, dashed, shift right] \arrow[rd, "\ker = \langle \alpha(P_B) + b\alpha(Q_B) \rangle"] & \\
 & & E_{AB} \\
\begin{array}{c} P_A, Q_A, \\ E_0, \\ P_B, Q_B \end{array} \arrow[ruu, "\ker \alpha = \langle A = P_A + aQ_A \rangle"] \arrow[rdd, "\ker \beta = \langle B = P_B + bQ_B \rangle"'] & \boxed{\begin{array}{c} E_A, \alpha(P_B), \alpha(Q_B), \\ E_B, \beta(P_A), \beta(Q_A) \end{array}} \arrow[uu, dashed, shift right] \arrow[dd, dashed, shift right] & \wr\| \\
 & & E_{BA} \\
 & E_B = \beta(E_0) \arrow[uu, dashed, shift right] \arrow[ru, "\ker = \langle \beta(P_A) + a\beta(Q_A) \rangle"'] &
\end{tikzcd}$$

Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

SIDH



BROKEN!

ode: Public, Alice’s secret, Bob’s secret

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.
- All isogeny-based schemes – Given elliptic curves E_0 and E_A, compute an isogeny $\alpha : E_0 \rightarrow E_A$ if it exists.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.
- All isogeny-based schemes – Given elliptic curves E_0 and E_A, compute an isogeny $\alpha : E_0 \rightarrow E_A$ if it exists.
- SIDH –

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \rightarrow E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

*Details if you care:

p a large prime; $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ supersingular; $\deg(\alpha)$, N public large smooth coprime integers; points P_B, Q_B chosen such that $\langle P_B, Q_B \rangle = E_0[N]$.

History of the SIDH problem

2011 Problem introduced by De Feo, Jao, and Plut

2016 Galbraith, Petit, Shani, Ti give active attack

2017 Petit gives passive attack on some parameter sets

2020 de Quehen, Kutas, Leonardi, M., Panny, Petit, Stange give passive attack on more parameter sets

2022 Castryck-Decru and Maino-M. give passive attack on SIKE parameter sets; Robert extends to all parameter sets

- CD and MM attack is subexponential in most cases
- CD attack polynomial-time when $\mathrm{End}(E_0)$ known
- Robert attack polynomial-time in all cases
- Panny and Pope implement MM attack; Wesolowski independently discovers direct recovery method

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α . (modulo technical restrictions)*

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α or $\ker(\alpha)$. (modulo technical restrictions)*

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α or $\ker(\alpha)$. (modulo technical restrictions)*

- * includes $E_0[N] = \langle P_B, Q_B \rangle$, so $E_A[N] = \langle \alpha(P_B), \alpha(Q_B) \rangle$.

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \rightarrow E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α or $\ker(\alpha)$. (modulo technical restrictions)*

- * includes $E_0[N] = \langle P_B, Q_B \rangle$, so $E_A[N] = \langle \alpha(P_B), \alpha(Q_B) \rangle$.
- If $\deg(\theta : E_A \rightarrow E_A) = N$, then $\ker(\theta) \subseteq E_A[N]$.

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \rightarrow E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α or $\ker(\alpha)$. (modulo technical restrictions)*

- * includes $E_0[N] = \langle P_B, Q_B \rangle$, so $E_A[N] = \langle \alpha(P_B), \alpha(Q_B) \rangle$.
- If $\deg(\theta : E_A \to E_A) = N$, then $\ker(\theta) \subseteq E_A[N]$.
- Computing $\ker(\widehat{\alpha}) \leftrightarrow$ computing $\ker(\alpha)$.

Technical interlude

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α or $\ker(\alpha)$. (modulo technical restrictions)*

- * includes $E_0[N] = \langle P_B, Q_B \rangle$, so $E_A[N] = \langle \alpha(P_B), \alpha(Q_B) \rangle$.
- If $\deg(\theta : E_A \to E_A) = N$, then $\ker(\theta) \subseteq E_A[N]$.
- Computing $\ker(\widehat{\alpha}) \leftrightarrow$ computing $\ker(\alpha)$.

$\rightsquigarrow$ Petit's idea: Construct $\theta : E_A \to E_A$ such that $\ker(\widehat{\alpha}) \subseteq \ker(\theta)$.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

$$\begin{tikzcd}
E_0 \arrow[r, bend left, "\alpha"] & E_A \arrow[l, bend left, "\widehat{\alpha}"]
\end{tikzcd}$$

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.

Petit’s trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha(E_0[N])$ (and $\widehat{\alpha}(E_A[N])$) from public torsion points.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha(E_0[N])$ (and $\widehat{\alpha}(E_A[N])$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha(E_0[N])$ (and $\widehat{\alpha}(E_A[N])$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.
- Restriction # 2: If there exist ι, n such that $\deg(\theta) = N$, then can completely determine θ, and α, in polynomial-time.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree.

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

- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha(E_0[N])$ (and $\widehat{\alpha}(E_A[N])$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.
- Restriction # 2: If there exist ι, n such that $\deg(\theta) = N$, then can completely determine θ, and α, in polynomial-time.
- Restriction # 2 rules out most interesting parameters, where $N \approx \deg(\alpha)$ (and $p \approx N \cdot \deg \alpha$).

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \rightarrow E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

Problem:

Not enough choices $\theta : E_A \to E_A$.

'No θ of degree N.'

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree N.'

Solution? $\theta : E_0 \times E_A \to E_0 \times E_A$?
$\rightsquigarrow$ still not enough.

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree N.'

Solution? $\theta : E_0 \times E_A \to E_0 \times E_A$?
$\rightsquigarrow$ still not enough. But!

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \rightarrow E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

Problem:
Not enough choices $\theta : E_A \rightarrow E_A$.
'No θ of degree N.'

Solution? $\theta : E_0 \times E_A \rightarrow E_0 \times E_A$?
$\rightsquigarrow$ still not enough. But! Kani's theorem:

- Constructs E_1, E_2 such that there exists a (polarisation-preserving) isogeny

$$E_1 \times E_A \rightarrow E_0 \times E_2$$

  of the right degree, N^2.
- Petit's trick then applies.

Recovering the secret

Finding the secret isogeny α of known degree.

$$\begin{tikzcd}
E_0 \arrow[loop left, "\iota"] \arrow[r, bend left, "\alpha"] & E_A \arrow[l, bend left, "\widehat{\alpha}"]
\end{tikzcd}$$

Recovering the secret

Finding the secret isogeny α of known degree.

$$\begin{tikzcd}
E_0 \arrow[r, bend left, "\alpha"] & E_A \\
E_1 \arrow[u, "\varphi"] &
\end{tikzcd}$$

Recovering the secret

Finding the secret isogeny α of known degree.

$$\begin{tikzcd}
E_0 \arrow[r, bend left, "\alpha"] & E_A \\
E_1 \arrow[u, dashed, "\varphi"] \arrow[ur, dashed, "f"] \arrow[r, dashed] & E_2 \arrow[u, dashed]
\end{tikzcd}$$

Kani's theorem constructs the above such that

$$\Phi = \begin{pmatrix} \varphi & -\widehat{\alpha} \\ * & * \end{pmatrix} : E_1 \times E_A \to E_0 \times E_2$$

is a polarisation-preserving isogeny of degree N^2, and

$$\ker(\Phi) = \{(\deg(\alpha)P, f(P)) : P \in E_1[N]\}$$

$\rightsquigarrow$ can compute Φ and read off secret α!

Recovering the secret with Robert's trick

Finding the secret isogeny α of known degree.

$$\begin{tikzcd}
{E_0}^4 \arrow[rr, bend left, "\alpha^4"] & & {E_A}^4 \\
E_0^4 \arrow[u, dashed, "\varphi"] \arrow[urr, dashed, "f"] \arrow[rr, dashed] & & E_A^4 \arrow[u, dashed]
\end{tikzcd}$$

constructs the above such that

$$\Phi = \begin{pmatrix} \varphi & -\widehat{\alpha}^4 \\ * & * \end{pmatrix} : E_0^4 \times E_A^4 \to E_0^4 \times E_A^4$$

is a polarisation-preserving isogeny of degree N^2, and

$$\ker(\Phi) \text{ is known}$$

$\rightsquigarrow$ can compute Φ and read off secret α!

What next?

- Fouotsa, Moriya, and Petit proposed mitigations
  - Masks either torsion point images or isogeny degrees
  - The mitigations make SIDH unusably slow and big
  - For advanced protocols may still be a good option
  - Cryptanalysis ongoing effort

What next?

- Fouotsa, Moriya, and Petit proposed mitigations
  - Masks either torsion point images or isogeny degrees
  - The mitigations make SIDH unusably slow and big
  - For advanced protocols may still be a good option
  - Cryptanalysis ongoing effort
- Constructive applications?
  - (Q)FESTA: New PKE. Fast and small as SIDH was?
  - SQISign2D: Small, fast, high security signatures.
  - polylog(N) storage/evaluation of $(N, \dots, N)$-isogenies$/\mathbb{F}_q$
  - poly-time algorithm for any instantiation of isogeny class-group action $/\mathbb{F}_q$
  - Work in progress with Maino and Robert $\rightsquigarrow$ computing genus 2 cyclic isogenies.

What next?

- Fouotsa, Moriya, and Petit proposed mitigations
  - Masks either torsion point images or isogeny degrees
  - The mitigations make SIDH unusably slow and big
  - For advanced protocols may still be a good option
  - Cryptanalysis ongoing effort
- Constructive applications?
  - (Q)FESTA: New PKE. Fast and small as SIDH was?
  - SQISign2D: Small, fast, high security signatures.
  - polylog(N) storage/evaluation of $(N, \ldots, N)$-isogenies$/\mathbb{F}_q$
  - poly-time algorithm for any instantiation of isogeny class-group action $/\mathbb{F}_q$
  - Work in progress with Maino and Robert $\rightsquigarrow$ computing genus 2 cyclic isogenies.

Thank you!