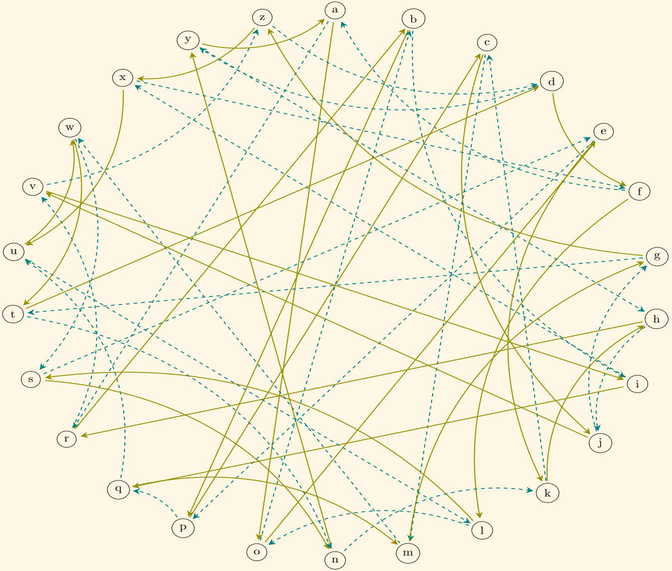
Applications of isogenies between abelian varieties to elliptic curves cryptosystems

2022/12/06 — VANTAGE seminar

Damien Robert

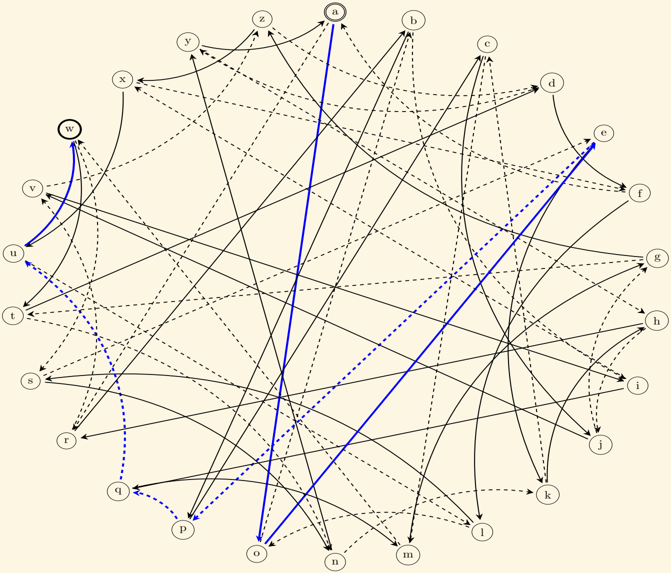
Équipe LFANT, Inria Bordeaux Sud-Ouest

Key exchange on a (commutative) graph



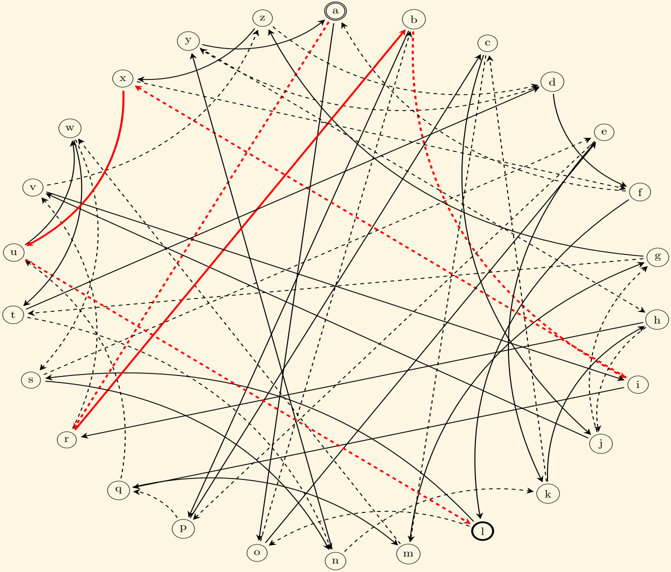
Key exchange on a (commutative) graph

Alice starts from 'a', follows the path 001110, and get 'w'.



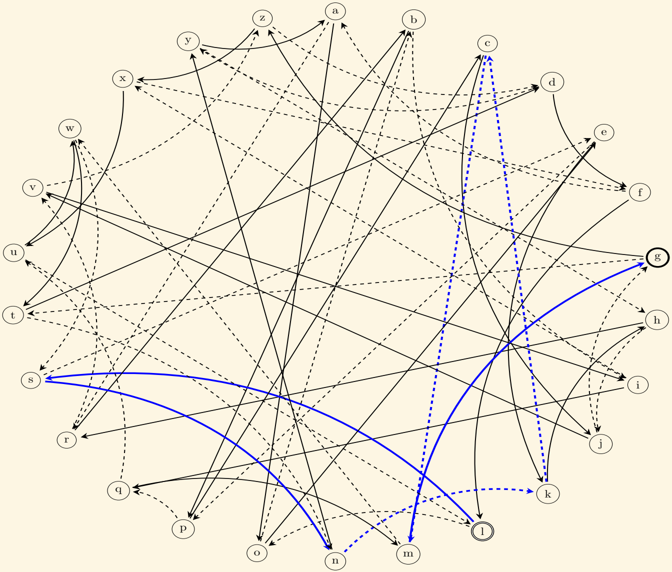
Key exchange on a (commutative) graph

Bob starts from ‘a’, follows the path 101101, and get ‘l’.



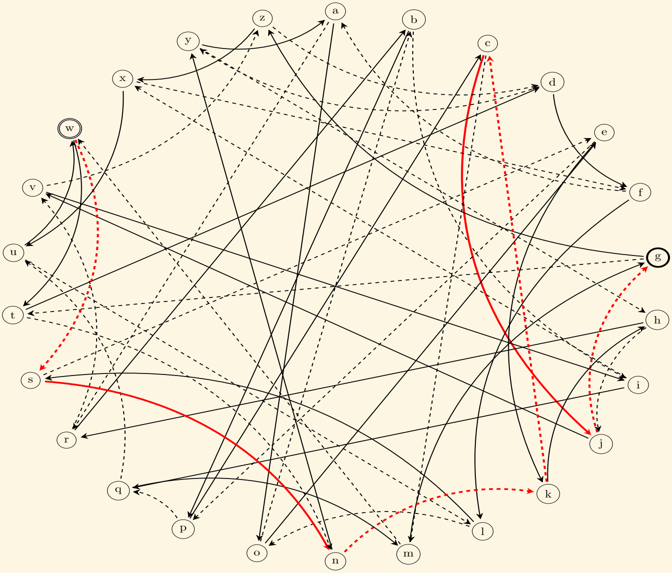
Key exchange on a (commutative) graph

Alice starts from ‘l’, follows the path 001110, and get ‘g’.



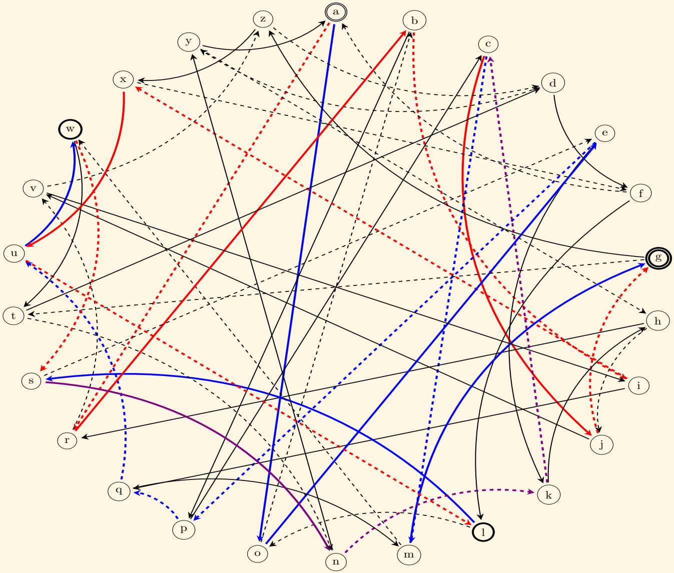
Key exchange on a (commutative) graph

Bob starts from 'w', follows the path 101101, and get 'g'.



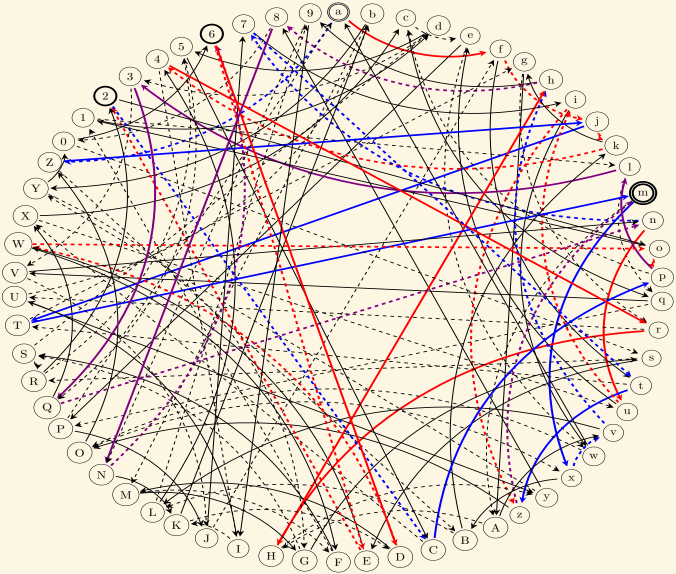
Key exchange on a (commutative) graph

The full exchange:



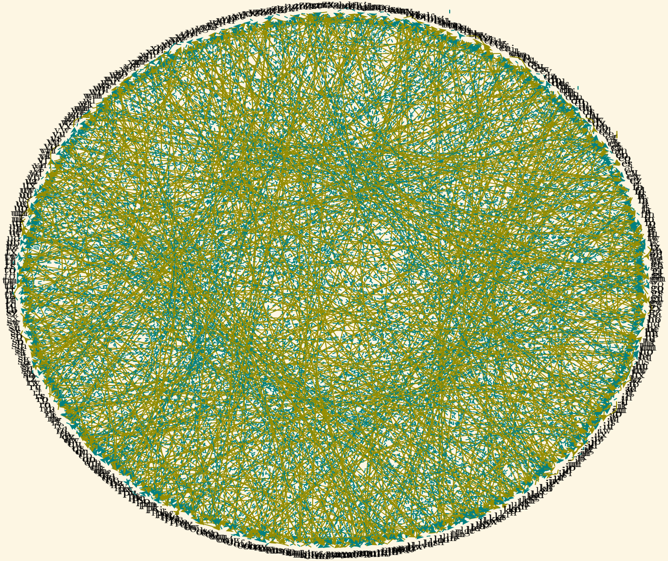
Key exchange on a (commutative) graph

Bigger graph (62 nodes)



Key exchange on a (commutative) graph

Even bigger graph (676 nodes)



Isogeny graphs for key exchange

- Needs a graph with good mixing properties:
  A path of length $O(\log N)$ gives a uniform node ⇒ Ramanujan/expander graph.
- The graph does not fit in memory.
- Needs an algorithm taking a node as input and giving the neighbour nodes as output.

- Isogeny graph of ordinary elliptic curves $E/\mathbb{F}_p$ [Couveignes (1997)], [Rostovtsev–Stolbunov (2006)]
- Graph of size $\approx \sqrt{p}$.
- Torsor (principal homogeneous space) under the class group $\mathrm{Cl}(\mathrm{End}(E_0))$.

☺ Commutative graph!

☹ Hidden shift problem solvable in quantum subexponential $L(1/2)$ time for an abelian group action via Kuperberg's algorithm.

- SIDH: supersingular elliptic curve Diffie-Helmann [De Feo, Jao (2011)],[De Feo, Jao, Plût (2014)]
- Use the isogeny graph of a supersingular elliptic curve E over $\mathbb{F}_{p^2}$.

Isogeny graphs for key exchange

SIDH in practice

- $p = 2^a 3^b - 1$. $N_A = 2^a, N_B = 3^b$, N_A prime to N_B.
- $E_0 : y^2 = x^3 + x$ (supersingular when $a \geq 2$) or $E_0 : y^2 = x^3 + 6x^2 + x$.
- $E_0[N_A] = \langle P_A, Q_A \rangle, E_0[N_B] = \langle P_B, Q_B \rangle$.
- Alice's secret isogeny: ϕ_A of kernel $\langle P_A + s_A Q_A \rangle$.
- Bob's secret isogeny: ϕ_B of kernel $\langle P_B + s_B Q_B \rangle$.
- Key exchange:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_B} & E_B \\ \downarrow \phi_A & & \downarrow \phi'_A \\ E_A & \xrightarrow{\phi'_B} & E_{AB} \end{array}$$

- E_{AB} is the shared secret.
- $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A : E_0 \to E_{AB}$ has kernel Ker ϕ_A + Ker ϕ_B.
- ϕ'_A has kernel $\langle \phi_B(P_A + s_A Q_A) \rangle$, ϕ'_B has kernel $\langle \phi_A(P_B + s_B Q_B) \rangle$.
- Alice publishes: $P'_B = \phi_A(P_B), Q'_B = \phi_A(Q_B)$.
  Bob publishes: $P'_A = \phi_B(P_A), Q'_A = \phi_B(Q_A)$. ("Torsion points".)
- Ker $\phi'_A = \langle P'_A + s_A Q'_A \rangle$, Ker $\phi'_B = \langle P'_B + s_B Q'_B \rangle$.
- Key exchange in $\widetilde{O}(\log N_A \ell_A^{1/2} + \log N_B \ell_B^{1/2})$
  (Via fast smooth isogeny computation [De Feo, Jao, Plût (2014)] and Velusqrt [Bernstein, De Feo, Leroux, Smith (2020)]).

SIDH in practice

- $p = 2^a 3^b - 1$. $N_A = 2^a, N_B = 3^b$, N_A prime to N_B.
- $E_0 : y^2 = x^3 + x$ (supersingular when $a \geq 2$) or $E_0 : y^2 = x^3 + 6x^2 + x$.
- $E_0[N_A] = \langle P_A, Q_A \rangle, E_0[N_B] = \langle P_B, Q_B \rangle$.
- Alice's secret isogeny: ϕ_A of kernel $\langle P_A + s_A Q_A \rangle$.
- Bob's secret isogeny: ϕ_B of kernel $\langle P_B + s_B Q_B \rangle$.
- Key exchange:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_B} & E_B \\ \downarrow{\scriptstyle \phi_A} & & \downarrow{\scriptstyle \phi'_A} \\ E_A & \xrightarrow{\phi'_B} & E_{AB} \end{array}$$

- E_{AB} is the shared secret.
- $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A : E_0 \to E_{AB}$ has kernel $\operatorname{Ker} \phi_A + \operatorname{Ker} \phi_B$.
- ϕ'_A has kernel $\langle \phi_B(P_A + s_A Q_A) \rangle$, ϕ'_B has kernel $\langle \phi_A(P_B + s_B Q_B) \rangle$.
- Alice publishes: $P'_B = \phi_A(P_B), Q'_B = \phi_A(Q_B)$.
  Bob publishes: $P'_A = \phi_B(P_A), Q'_A = \phi_B(Q_A)$. ("Torsion points".)
- $\operatorname{Ker} \phi'_A = \langle P'_A + s_A Q'_A \rangle$, $\operatorname{Ker} \phi'_B = \langle P'_B + s_B Q'_B \rangle$.
- Key exchange in $\widetilde{O}(\log N_A \ell_A^{1/2} + \log N_B \ell_B^{1/2})$
  (Via fast smooth isogeny computation [De Feo, Jao, Plût (2014)] and Velusqrt [Bernstein, De Feo, Leroux, Smith (2020)]).

SIDH in practice

- $p = 2^a 3^b - 1$. $N_A = 2^a, N_B = 3^b$, N_A prime to N_B.
- $E_0 : y^2 = x^3 + x$ (supersingular when $a \geq 2$) or $E_0 : y^2 = x^3 + 6x^2 + x$.
- $E_0[N_A] = \langle P_A, Q_A \rangle, E_0[N_B] = \langle P_B, Q_B \rangle$.
- Alice's secret isogeny: ϕ_A of kernel $\langle P_A + s_A Q_A \rangle$.
- Bob's secret isogeny: ϕ_B of kernel $\langle P_B + s_B Q_B \rangle$.
- Key exchange:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_B} & E_B \\ \downarrow{\scriptstyle \phi_A} & & \downarrow{\scriptstyle \phi'_A} \\ E_A & \xrightarrow{\phi'_B} & E_{AB} \end{array}$$

- E_{AB} is the shared secret.
- $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A : E_0 \to E_{AB}$ has kernel $\operatorname{Ker} \phi_A + \operatorname{Ker} \phi_B$.
- ϕ'_A has kernel $\langle \phi_B(P_A + s_A Q_A) \rangle$, ϕ'_B has kernel $\langle \phi_A(P_B + s_B Q_B) \rangle$.
- Alice publishes: $P'_B = \phi_A(P_B), Q'_B = \phi_A(Q_B)$.
  Bob publishes: $P'_A = \phi_B(P_A), Q'_A = \phi_B(Q_A)$. ("Torsion points".)
- $\operatorname{Ker} \phi'_A = \langle P'_A + s_A Q'_A \rangle$, $\operatorname{Ker} \phi'_B = \langle P'_B + s_B Q'_B \rangle$.
- Key exchange in $\widetilde{O}(\log N_A \ell_A^{1/2} + \log N_B \ell_B^{1/2})$
  (Via fast smooth isogeny computation [De Feo, Jao, Plût (2014)] and Velusqrt [Bernstein, De Feo, Leroux, Smith (2020)]).

SIDH in practice

- $p = 2^a 3^b - 1$. $N_A = 2^a, N_B = 3^b$, N_A prime to N_B.
- $E_0 : y^2 = x^3 + x$ (supersingular when $a \geq 2$) or $E_0 : y^2 = x^3 + 6x^2 + x$.
- $E_0[N_A] = \langle P_A, Q_A \rangle, E_0[N_B] = \langle P_B, Q_B \rangle$.
- Alice's secret isogeny: ϕ_A of kernel $\langle P_A + s_A Q_A \rangle$.
- Bob's secret isogeny: ϕ_B of kernel $\langle P_B + s_B Q_B \rangle$.
- Key exchange:

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_B} & E_B \\ \downarrow{\scriptstyle \phi_A} & & \downarrow{\scriptstyle \phi'_A} \\ E_A & \xrightarrow{\phi'_B} & E_{AB} \end{array}$$

- E_{AB} is the shared secret.
- $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A : E_0 \to E_{AB}$ has kernel $\operatorname{Ker} \phi_A + \operatorname{Ker} \phi_B$.
- ϕ'_A has kernel $\langle \phi_B(P_A + s_A Q_A) \rangle$, ϕ'_B has kernel $\langle \phi_A(P_B + s_B Q_B) \rangle$.
- Alice publishes: $P'_B = \phi_A(P_B), Q'_B = \phi_A(Q_B)$.
  Bob publishes: $P'_A = \phi_B(P_A), Q'_A = \phi_B(Q_A)$. ("Torsion points".)
- $\operatorname{Ker} \phi'_A = \langle P'_A + s_A Q'_A \rangle$, $\operatorname{Ker} \phi'_B = \langle P'_B + s_B Q'_B \rangle$.
- Key exchange in $\widetilde{O}(\log N_A \ell_A^{1/2} + \log N_B \ell_B^{1/2})$
  (Via fast smooth isogeny computation [De Feo, Jao, Plût (2014)] and Velusqrt [Bernstein, De Feo, Leroux, Smith (2020)]).

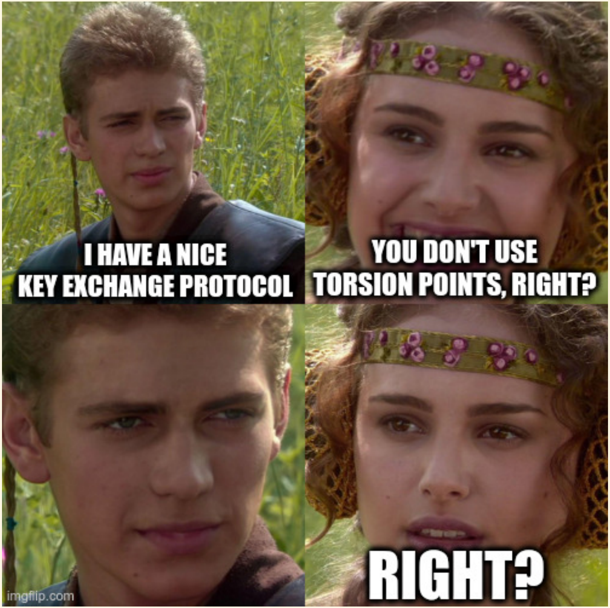
Isogeny evaluation and interpolation

- Evaluation: given an N-isogeny f and a point $Q \in E(\mathbb{F}_q)$, evaluate $f(Q)$.
- N-evaluation problem: f is an N-isogeny = Ker f is of degree N.

- Interpolation: given a tuple $(P, f(P))$, recover f.
- (N, N')-interpolation problem: given f an N-isogeny and P a point of N'-torsion, from $(P, f(P))$ and $Q \in E(\mathbb{F}_q)$, evaluate $f(Q)$ ($N' \geq N$).
- Weak interpolation: we are given $(P_1, f(P_1))$, $(P_2, f(P_2))$ for (P_1, P_2) a basis of $E[N]$.

- SIDH: the key exchange uses the N_A and N_B evaluation problems
- If we can solve the weak interpolation problem when $N = N_A, N' = N_B$ are smooths in polylogarithmic time, we can break SIDH.

Isogeny evaluation and interpolation



Evaluation

- $f(x,y) = \left(\frac{g(x)}{h(x)}, cy\left(\frac{g(x)}{h(x)}\right)'\right)$;
- [Vélu]: given the kernel $\operatorname{Ker} f : \{P \in E \mid h(x(P)) = 0\}$ of degree N, can evaluate $f(Q)$ in $O(N)$ operations in $\mathbb{F}_q$.
- Velusqrt: in the special case $\operatorname{Ker} f = \langle T \rangle, T \in \mathbb{F}_q$, can evaluate $f(Q)$ in $\widetilde{O}(\sqrt{N})$ operations in $\mathbb{F}_q$.
- Linear time.

- If N is smooth, f can be decomposed into a product of small isogenies.
- Evaluation in $O(\log N \ell_N)$ or $\widetilde{O}(\log N \sqrt{\ell_N})$.
- Logarithmic time.

- The decomposition cost is quasi-logarithmic if $\operatorname{Ker} f = \langle T \rangle$ with $T \in \mathbb{F}_q$; polylogarithmic if N' is powersmooth; but linear if T lives in a large extension.

Interpolation

- Given $(P, f(P))$, P a point of order $N' \geq 2N$, we can recover the rational function $\frac{g(x)}{h(x)}$ in $\widetilde{O}(N)$ by interpolating the points $(x(mP), x(mf(P)))$, $m = 1, \ldots, N' - 1$.
- Can evaluate on Q directly.
- Special case when $p > 2N$: $P \neq 0 \in T_{0_E}(E)$, a "fat point" of order $p \Rightarrow$ solve a differential equation [Elkies].
- Quasi-linear time.

- Faster algorithm when N' is smooth?
- Yes if $f(P) = 0$. Then $N = N'$ and $\mathrm{Ker} f = \langle P \rangle$.
- If $N = N'$, the weak interpolation problem reduces via the DLP to the N-evaluation problem.
- This is why the SIDH key exchange is fast: Bob uses the torsion point information published by Alice to find the kernel of his pushforward isogeny.
- No reason to expect a fast algorithm when N' is prime to N.

Interpolation

- Given $(P, f(P))$, P a point of order $N' \geq 2N$, we can recover the rational function $\frac{g(x)}{h(x)}$ in $\widetilde{O}(N)$ by interpolating the points $(x(mP), x(mf(P)))$, $m = 1, \ldots, N'-1$.
- Can evaluate on Q directly.
- Special case when $p > 2N$: $P \neq 0 \in T_{0_E}(E)$, a "fat point" of order $p \Rightarrow$ solve a differential equation [Elkies].
- Quasi-linear time.

- Faster algorithm when N' is smooth?
- Yes if $f(P) = 0$. Then $N = N'$ and $\mathrm{Ker}\, f = \langle P \rangle$.
- If $N = N'$, the weak interpolation problem reduces via the DLP to the N-evaluation problem.
- This is why the SIDH key exchange is fast: Bob uses the torsion point information published by Alice to find the kernel of his pushforward isogeny.
- ~~No reason to expect a fast algorithm when N' is prime to N.~~

Revisiting isogeny evaluation

- Can an N-isogeny be evaluated faster than linear time when N has a large prime factor?
- If $f = [\ell]$ (so $N = \ell^2$): double and add in $O(\log \ell)$ to evaluate ℓQ.

- $F : E^2 \to E^2, (P_1, P_2) \mapsto (P_1 + P_2, P_1 - P_2)$ is a 2-isogeny in dimension 2.
- Double: $F(P, P) = (2P, 0)$.
- Add: $F(P, Q) = (P + Q, P - Q)$.

- We can evaluate ℓQ as a composition of $O(\log \ell)$ evaluations of F, projections $E^2 \to E$ and embeddings $E \to E^2$.
- Double and add on E = 2-isogenies in dimension 2

Polarisations on an abelian variety

If A is an abelian variety, a polarisation is:

- a (symmetric) isogeny $\lambda_A : A \to \widehat{A}$;
- an (algebraic equivalence class) of an ample divisor Θ_A;
- an (anti-symmetric) pairing $T_\ell(A) \times T_\ell(A) \to \mathbb{G}_m$;
- projective coordinates $A \dashrightarrow \mathbb{P}^m_k$ (up to translation)

Principal polarisation= λ_A is an isomorphism: principally polarized abelian variety (ppav)

N-isogenies

- $f : (A, \lambda_A) \to (B, \lambda_B)$ is an N-isogeny between ppav if $f^* \lambda_B = N \lambda_A$.
- Dual isogeny: $\hat{f} : \hat{B} \to \widehat{A}$
- Contragredient isogeny / Dual with respect to the principal polarisations:

$$\tilde{f} = \lambda_A^{-1} \hat{f} \lambda_B : B \to A$$

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle \lambda_A^{-1}} \uparrow & & \downarrow {\scriptstyle \lambda_B} \\ \widehat{A} & \xleftarrow[\hat{f}]{} & \hat{B} \end{array}$$

- f is an N-isogeny $\Leftrightarrow \tilde{f} f = N \Leftrightarrow f \tilde{f} = N$.
- $\operatorname{Ker} f = \operatorname{Im} \left(\tilde{f} \mid B[N] \right)$.

N-isogenies and isotropic kernels

- $f : (A, \lambda_A) \to (B, \lambda_B)$ N-isogeny $\Rightarrow$ Ker f is maximal isotropic in $A[N]$ for the Weil pairing
- Conversely, if $K \subset A[N]$ maximal isotropic, $N\lambda_A$ descends to a principal polarisation on $B = A/K$.

- An elliptic curve only has one principal polarisation ($NS(E) = \mathbb{Z}$).
- So $f : E_1 \to E_2$ is an N-isogeny $\Leftrightarrow$ # Ker $f = N$.
- But in higher dimension there may be many non equivalent principal polarisations.

Example (Superspecial abelian surfaces)

$A = E^2, E/\mathbb{F}_{p^2}$ supersingular. It admits $\approx p^2/288$ product polarisations $(E_1 \times E_2, \lambda_{E_1} \times \lambda_{E_2})$ where E_1, E_2 are supersingular and $\approx p^3/2880$ indecomposable polarisations (Jac C, Θ_C) where C is an hyperelliptic curve of genus 2.

- If $f : (A, \lambda_A) \to (B, \lambda_B)$ has maximal isotropic kernel in $A[N]$, $N\lambda_A$ descends to a principal polarisation λ'_B on B.
- But we may have $\lambda'_B \neq \lambda_B$.
- $\tilde{f} \circ f = N$ is a stronger condition that ensures compatibility of f with λ_B.

Algorithms for N-isogenies

- [Cosset-R. (2014), Lubicz-R. (2012–2022)]: An N-isogeny in dimension g can be evaluated in linear time $O(N^g)$ arithmetic operations in the theta model given generators of its kernel.
- Warning: exponential dependency 2^g or 4^g in the dimension g.

- [Couveignes-Ezome (2015)]: Algorithm in $O(N^g)$ in the Jacobian model.
- Not hard to extend to product of Jacobians.
- Restricted to $g \leq 3$.

Composition and product polarisations

- Composition: $f : A \to B$ a N-isogeny, $g : B \to C$ a M-isogeny, $g \circ f : A \to C$.
- $\widehat{g \circ f} = \hat{f} \circ \hat{g} : \hat{C} \to \hat{A}$;
- $\widetilde{g \circ f} = \tilde{f} \circ \tilde{g} : C \to A$;
- $(\widetilde{g \circ f}) \circ (g \circ f) = \tilde{f} \circ \tilde{g} \circ g \circ f = NM$.
- The composition $g \circ f$ is an NM-isogeny.
- Conversely, if $g \circ f$ is an N-isogeny and f (resp. g) is an M-isogeny, then g (resp. f) is an N/M-isogeny.

- Product polarisation: $(A, \lambda_A) \times (B, \lambda_B) = (A \times B, \lambda_A \times \lambda_B)$ where $\lambda_A \times \lambda_B : A \times B \to \widehat{A} \times \widehat{B}$ is the product.

- $F = \begin{pmatrix} a & c \\ b & d \end{pmatrix} : (A \times B, \lambda_A \times \lambda_B) \to (C \times D, \lambda_C \times \lambda_D)$.
- $\hat{F} = \begin{pmatrix} \hat{a} & \hat{b} \\ \hat{c} & \hat{d} \end{pmatrix} : \hat{C} \times \hat{D} \to \hat{A} \times \hat{B}$.
- $\tilde{F} = \begin{pmatrix} \tilde{a} & \tilde{b} \\ \tilde{c} & \tilde{d} \end{pmatrix} : C \times D \to A \times B$.

Kani's lemma [Kani (1997)], [R. (2022-08)]

- $\alpha : A \to B$ a a-isogeny, $\beta : A \to C$ a b-isogeny.
- $\alpha' : C \to D$ a a-isogeny, $\beta' : C \to D$ a b-isogeny with $\beta'\alpha = \alpha'\beta$:

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ \downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\beta'} \\ C & \xrightarrow{\alpha'} & D \end{array}$$

- NB: If a prime to b, the pushforward α', β' of α, β by β, α satisfy these conditions.

- $F = \begin{pmatrix} \alpha & \widetilde{\beta'} \\ -\beta & \widetilde{\alpha'} \end{pmatrix} : A \times D \to B \times C.$
- $\tilde{F} = \begin{pmatrix} \tilde{\alpha} & -\tilde{\beta} \\ \beta' & \alpha' \end{pmatrix} : B \times C \to A \times D, \quad \tilde{F}F = a + b.$
- F is an $a + b$-isogeny with respect to the product polarisations.
- Ker $F = \{\tilde{\alpha}(P), \beta'(P) \mid P \in B[a + b]\}$ (if a is prime to b)

Revisiting the interpolation

- If we know $f(E[N'])$, and we can find a $m = N' - N$ isogeny α that we can evaluate on $E[N']$, we recover $\operatorname{Ker} F$.
- We can then evaluate F, hence f at any point: $F(P,0) = (\alpha(P), -f(P)) = F(P,0)$.
- This evaluation is fast if N' is smooth.

Examples:

- m smooth [Maino-Martindale]
- $m = \ell^2$: take $\alpha = [\ell]$;
- $\operatorname{End}(E)$ has an efficient endomorphism of norm m [Castryck-Decru].

The general case

- $\alpha = \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$ is always an endomorphism of norm $a_1^2 + a_2^2$ on E^2 (Gaussian integers $\mathbb{Z}[i]$);
- $\alpha = \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix}$ is always an endomorphism of norm $m = a_1^2 + a_2^2 + a_3^2 + a_4^2$ on E^4 (Hamilton's quaternion algebra)
- Evaluating α costs $O(\log m)$ arithmetic operations;
- Every integer is a sum of four squares [Διόφαντος ὁ Ἀλεξανδρεύς, Lagrange].

The embedding lemma [R.]

- A N-isogeny $f : A \to B$ in dimension g can always be efficiently embedded into a N' isogeny $F : A' \to B'$ in dimension $8g$ (and sometimes $4g, 2g$) for any $N' \geq N$.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow & & \uparrow \\ A' & \xrightarrow{F} & B' \end{array}$$

- Considerable flexibility (at the cost of going up in dimension).

- Breaks SIDH ([Castryck-Decru], [Maino-Martindale] in dimension 2, [R.] in dimension 4 or 8)
- Reduces the (N, N')-weak interpolation problem to the N'-evaluation problem in higher dimension;
- Only needs $N'^2 \geq N$ (uses the dual isogeny)

⇒ Solves the weak interpolation problem when N' is (power) smooth

- Amazing fact: does not requires $\operatorname{Ker} f$, works even if N is prime
- Open question: case N' prime? Can we find a fast N'-evaluation algorithm?

Efficient representation of isogenies [R.]

- For the N-evaluation problem, once we have evaluated f on a basis of the N'-torsion this reduces to the N'-weak interpolation problem which reduces to the N'-evaluation problem (in higher dimension).

- Can always embed an N-isogeny f into a N'-isogeny with N' powersmooth;
- Then decompose F as a product of small isogenies: polylogarithmic space $O(\log^3 N)$;
- We need to evaluate f on the N'-torsion: decomposition is quasi-linear;

- Evaluation in polylogarithmic time $O(\log^7 N)$ arithmetic operations.

Point counting

- The Frobenius π_p can be evaluated in $O(\log p)$ arithmetic operations;
- Its action on the tangent space $T_{0_E}E$ is trivial ☹;
- The action $\lambda \mod p$ of the Verschiebung $\widetilde{\pi}_p$ on $T_{0_E}E$ is non trivial (if E is ordinary), and gives the trace $t = \lambda + q/\lambda$ of π_p modulo p ☺;
- Since $\widetilde{\pi}_p \circ \pi_p = [p]$, the Verschiebung can be efficiently evaluated on the image of π_p ☺;
- But $\pi_p(T_{0_E}E) = 0$ ☹.

- We can instead embed π_p (and $\widetilde{\pi}_p$) into a powersmooth separable isogeny F and evaluate F on the tangent space!
- Polynomial point counting algorithm: $\lambda \mod p$ in $O(\log^{10} p)$ arithmetic operations.
- Similar to Schoof's algorithm (but slower): evaluate π_p on small ℓ_i-torsion points.
- Rather than doing a DLP on these points to reconstruct $t \mod \prod \ell_i$, we reconstruct a $\prod \ell_i$-isogeny F embedding the Frobenius.

- A lift of F gives a lift of π_p. So we can compute the action of π_p on the deformation space of E.

⇒ Compute canonical lift $\widetilde{E}$ in time polynomial in $O(\log p)$!

Point counting and canonical lifts

$E/\mathbb{F}_q, q = p^n$.

- [Schoof 1985]: $\widetilde{O}(\log^5 q) = \widetilde{O}(n^5 \log^5 p)$ (Étale cohomology)
- [SEA 1992]: $\widetilde{O}(\log^4 q) = \widetilde{O}(n^4 \log^4 p)$

- [Kedlaya 2001]: $\widetilde{O}(n^3 p)$ (Rigid cohomology)
- [Harvey 2007]: $\widetilde{O}(n^{3.5} p^{1/2} + n^5 \log p)$

- [Satoh 2000] (canonical lifts of ordinary curves): $\widetilde{O}(n^2 p^2)$ (Crystalline cohomology)
- [Maiga – R. 2021]: $\widetilde{O}(n^2 p)$
- [R. 2022]: $\widetilde{O}(n^2 \log^8 p + n \log^{11} p)$