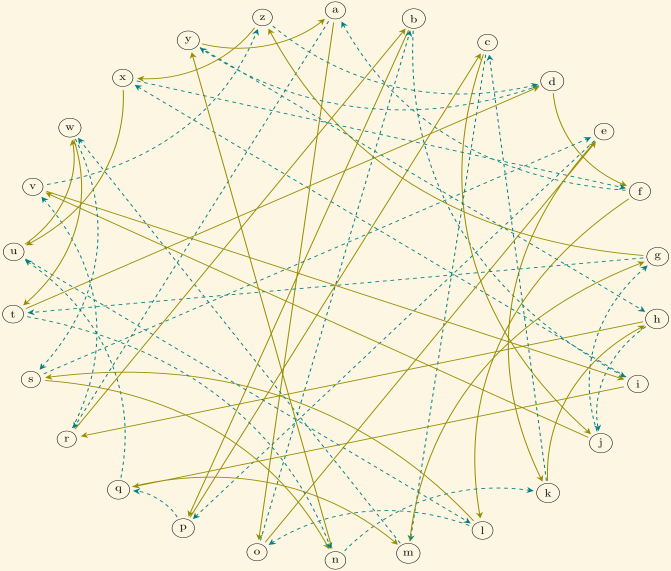
New applications of higher dimensional isogenies

2023/09/21 — Loria, Nancy

Damien Robert

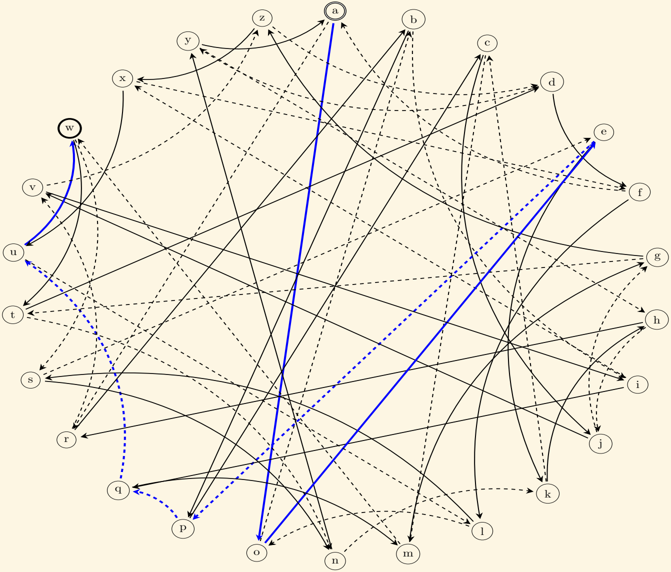
Équipe Canari, Inria Bordeaux Sud-Ouest

Key exchange on a (commutative) graph



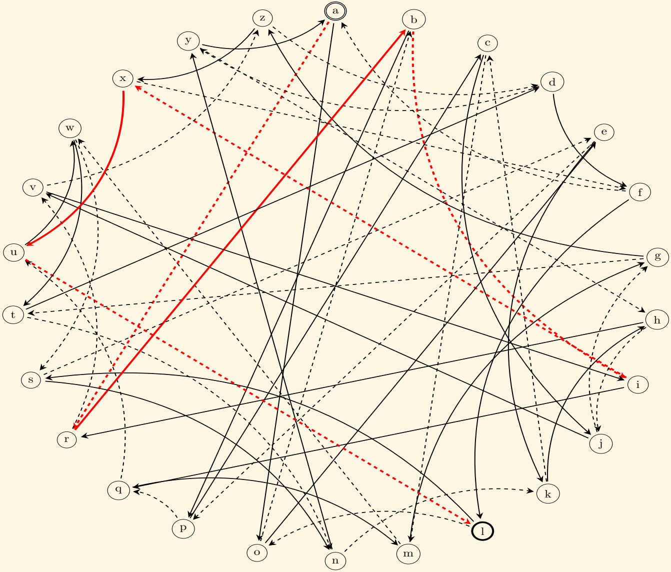
Key exchange on a (commutative) graph

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



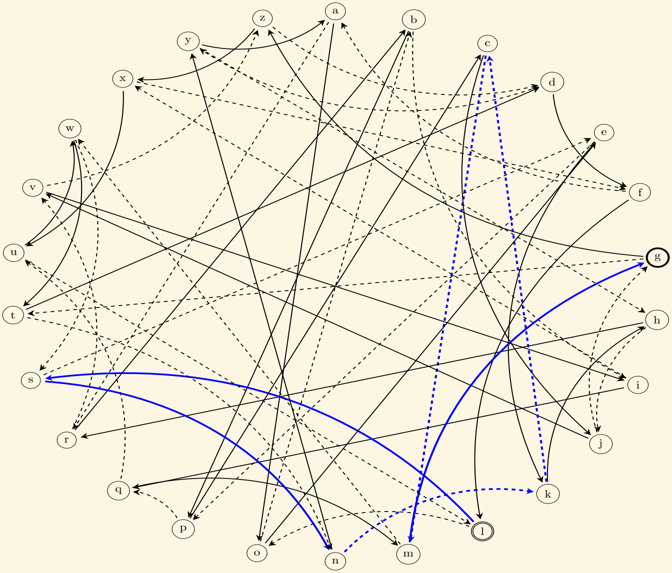
Key exchange on a (commutative) graph

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



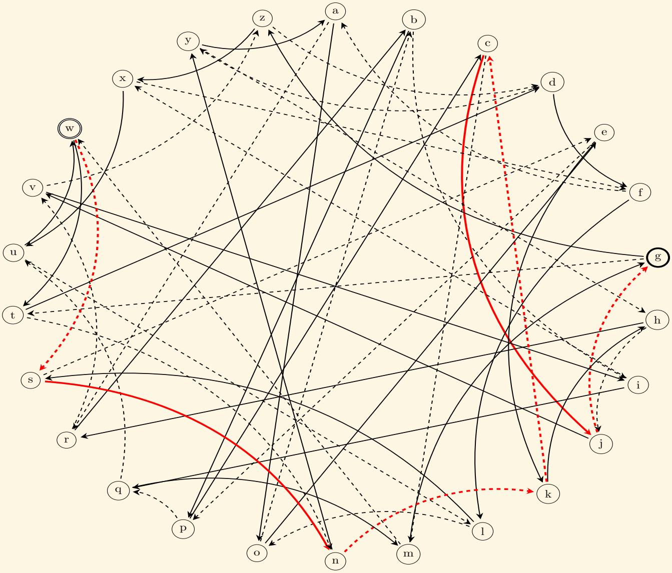
Key exchange on a (commutative) graph

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



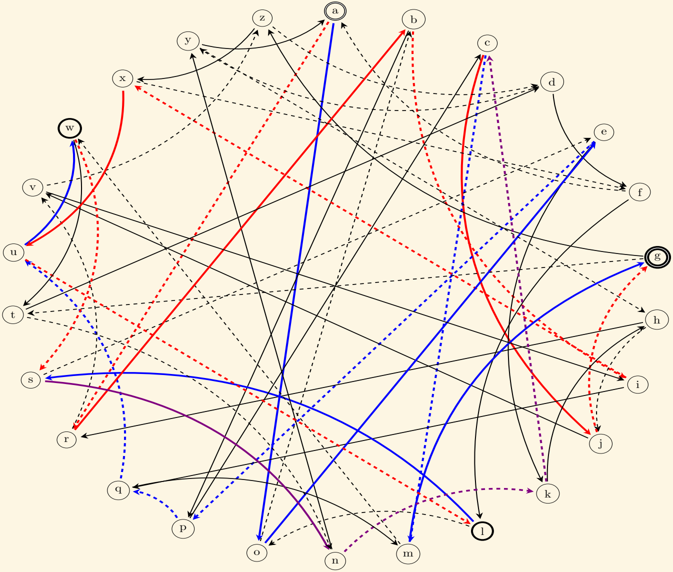
Key exchange on a (commutative) graph

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



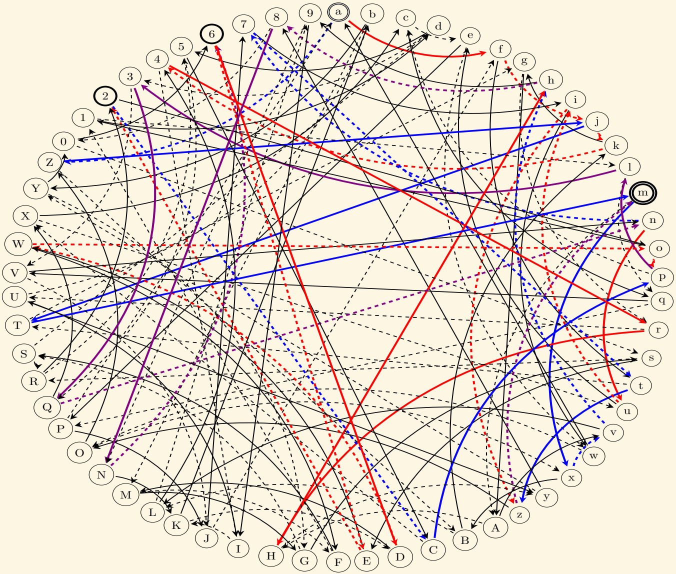
Key exchange on a (commutative) graph

The full exchange:



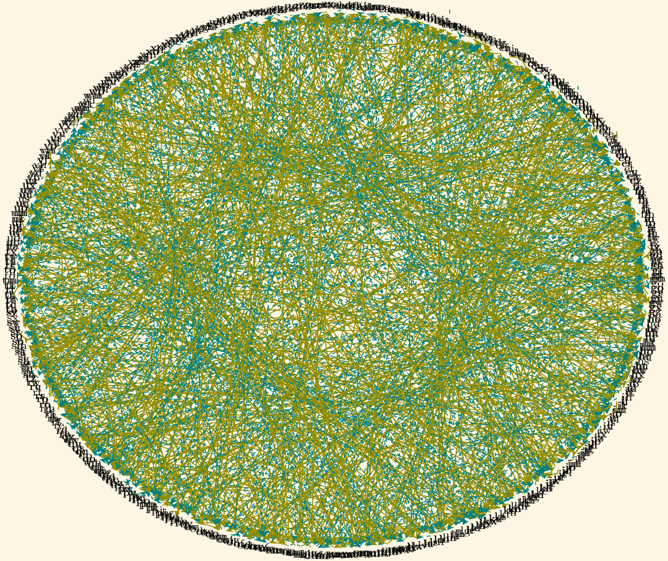
Key exchange on a (commutative) graph

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 $(N = 2^{256})$.
- 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 $N \approx \sqrt{p}$.
- 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 $\mathbb{F}_{p^2}$ $(N \approx p)$.

Isogeny graphs for key exchange

Meme: Gru's plan

- Isogeny based key exchange
- Use supersingular curves
- The graph is non commutative
- The graph is non commutative

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 $a \geq 2$)
- $E_0[N_A] = \langle P_A, Q_A \rangle, E_0[N_B] = \langle P_B, Q_B \rangle$.
- 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.
- $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A : E_0 \rightarrow E_{AB}$ 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 $\widetilde{O}(\log N_A \ell_A + \log N_B \ell_B)$
  (Via fast smooth isogeny computation [De Feo, Jao, Plût (2014)])

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 = $\operatorname{Ker} f$ 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 smooth in polylogarithmic time, we can break SIDH.

Meme: Anakin

- I have a nice key exchange protocol
- You don't use torsion points, right?
- …
- Right?

Evaluation

- $f : E_1 \to E_2$ an N-isogeny
- $f(x,y) = \left(\frac{g(x)}{h(x)}, cy\left(\frac{g(x)}{h(x)}\right)'\right)$, $\deg g, \deg h \leq N$
- [Vélu 1971]: given $h(x)$ representing the kernel $\operatorname{Ker} f : \{P \in E \mid h(x(P)) = 0\}$, evaluate $f(Q)$ in $O(N)$ operations in $\mathbb{F}_q$.
- Kernel representation: Linear time and linear space.

- Velusqrt special case $\operatorname{Ker} f = \langle T \rangle, T \in \mathbb{F}_{q^d}$, evaluate $f(Q)$ in $\widetilde{O}(\sqrt{N})$ operations in $\mathbb{F}_{q^d}$.
- Generator representation: Compact representation if d small.

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

- The decomposition cost is quasi-logarithmic if $\operatorname{Ker} f = \langle T \rangle$ with $T \in \mathbb{F}_q$ (or lives in a small extension); hence polylogarithmic if N is powersmooth; but linear if T lives in a large extension.
- In SIDH: the A and B torsion points are rational, so the decomposition is fast!

Interpolation

- Given $(P, f(P))$, P a point of order $N' \geq 2N$, 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.
- Quasi-linear time.

- Faster algorithm when N' is smooth?
- Yes if $f(P) = 0$. Then $N = N'$ and $\operatorname{Ker} f = \langle P \rangle$.
- 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 $N' \geq 2N$, 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.
- Quasi-linear time.

- Faster algorithm when N' is smooth?
- Yes if $f(P) = 0$. Then $N = N'$ and $\operatorname{Ker} f = \langle P \rangle$.
- 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.
- $F = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$
- Double: $F(T, T) = (2T, 0)$.
- Add: $F(T, Q) = (T + Q, T - Q)$.

- 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

Kani's lemma [Kani 1997] ($g = 1$), [R. 2022] ($g > 1$)

- $\alpha : A \to B$ 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 $\beta'\alpha = \alpha'\beta$:

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

- If a prime to b, the pushforward α', β' of α by β 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.

- $\operatorname{Ker} F = \{\tilde{\alpha}(P), \beta'(P) \mid P \in B[a + b]\}$ (if a is prime to b)

Using Kani's lemma for the interpolation

$$\begin{array}{ccc} E_1 & \xrightarrow{f} & E_2 \\ \downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \alpha'} \\ E_1' & \xrightarrow{f'} & E_2' \end{array}$$

- $f : E_1 \to E_2$ an N-isogeny.
- Goal: replace f by F an N'-isogeny.
- Find $\alpha : E_1 \to E_1'$ an m-isogeny, with $N' = N + m$.
- Kani's lemma: $F : E_1 \times E_2' \to E_1' \times E_2$ is an N'-isogeny.
- We know $f(E[N'])$ and we can evaluate α on $E[N'] \Rightarrow$ recover $\mathrm{Ker}\, F$ (or $\mathrm{Ker}\, \tilde{F}$)
- Evaluate F, hence f at any point: $F(P, 0) = (\alpha(P), -f(P))$.
- Evaluation is fast if N' is (power) smooth.

Examples:

- m smooth [Castryck–Decru; Maino–Martindale (2022)]
- $m = \ell^2$: take $\alpha = [\ell]$
- $\mathrm{End}(E)$ has an efficient endomorphism α of norm m [Castryck–Decru; Wesolowski (2022)].

The general case: Zahrin's trick

- $\alpha = \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix}$ is always an endomorphism of norm $m = a_1^2 + a_2^2$ 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: $O(\log m)$ arithmetic operations
- Every integer is a sum of four squares.

$$\begin{array}{ccc} E_1^4 & \xrightarrow{f} & E_2^4 \\ \downarrow \alpha & & \downarrow \alpha \\ E_1^4 & \xrightarrow{f} & E_2^4 \end{array}$$

- $F : E_1^4 \times E_2^4 \to E_1^4 \times E_2^4$ is an N'-isogeny.

Kani's lemma + Zahrin's trick = the embedding lemma [R. 2022]

- 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).

- Reduces the weak (N, N')-interpolation problem to the N'-evaluation problem in higher dimension
- Actually only need the image of f on a subgroup of size N', $N' > 4N$ (via further tricks by Castryck, De Feo, R., Wesolowski…)

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

- Amazing fact: does not requires $\operatorname{Ker} f$, works even if N is prime

- Breaks SIDH: [Castryck–Decru], [Maino–Martindale] in dimension 2, [R.] in dimension 4 or 8

Kani's lemma + Zahrin's trick = the embedding lemma [R. 2022]

Meme: disaster girl

- SIDH
- Higher dimensional isogenies

Efficient representation of isogenies [R. 2022]

- If we know the evaluation of f on a basis of $E[N']$, we can replace f by a N'-isogeny F in higher dimension

⇒ Polylogarithmic time and space

- Previously: linear time (for a general isogeny)

- Torsion representation: $(P_i, Q_i, f(P_i), f(Q_i))$ for (P_i, Q_i) basis of $E[\ell_i^{e_i}]$, small torsion points ($N' = \prod \ell_i^{e_i} > 2\sqrt{N}$)
- If $E[2^n] = \langle P_1, P_2 \rangle$ is rational, take $N' = 2^n$.

- The torsion representation is an universal efficient representation
- We just need the image of f on enough nice points
- Corollary: If f has an efficient representation, so does f/m (division) and $\hat{f}$ (dual)

Some algorithmic applications [R. 2022]

- $E/\mathbb{F}_q$ ordinary elliptic curve, $K = \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$. Given the factorisation of $[O_K : \mathbb{Z}[\pi]]$, compute $\mathrm{End}(E)$ in polynomial time.
  Factorisation: quantum polynomial time, classical subexponential time
- Previously: no quantum polynomial time algorithm known.
  Classical algorithm in $L(1/2)$ under GRH [Bisson–Sutherland 2009].

- Compute the canonical lift $\hat{E}/\mathbb{Z}_q$ in polynomial time.
- Previously: $L(1/2)$ under GRH [Couveignes–Henocq 2002]

- Compute the modular polynomial Φ_ℓ in quasi-linear time in any dimension g.
- Previously: no algorithm known to compute Φ_ℓ in quasi-linear time when $g > 2$.

Point counting and canonical lifts

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

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

- [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]: $\widetilde{O}(n^2 \log^8 p + n \log^{11} p)$

Cryptographic applications

- Free protocols from the shackle of using only smooth degree isogenies
- Choose E with large rational 2^m-torsion $\Rightarrow$ embed N-isogenies into higher dimensional 2^m-isogenies

- SQISignHD [Dartois, Leroux, R., Wesolowski 2023]: post-quantum signature scheme
- Signing in dimension 1, verification in dimension 4
- Public key: 64B, Signature: 105B
  Prior Art: SQISign: 204B, Lattices: 666B–2420B, (ECDSA: 64B)

- FESTA [Basso, Maino, Pope 2023]: encryption in dimension 1, decryption in dimension 2 (or 4)
- VRF [Leroux 2023]: use dimension up to 4
  Partial VDF construction by [Maino 2023]: use dimension 2
- Identity based encryption [Fouotsa 2023]: use dimension 8

Cryptographic applications

Meme: Buzz

- Higher dimensional isogenies
- Higher dimensional isogenies everywhere

Algorithms for N-isogenies in higher dimension

- [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 $g \leq 3$.

2^m-isogenies in higher dimension

- [R. 2023]: faster formula for 2^m-isogenies in the theta model
- Decomposition: g points to push, 2^g coordinates by point
- Cost compared to dimension 1: dimension 2: $\times 4$, dimension 4: $\times 32$, dimension 8: $\times 1024$.
- Images: dimension 2: $\times 2$, dimension 4: $\times 8$, dimension 8: $\times 128$.

- Optimised Sage implementation of 2^m-isogenies in dimension 2 (with Dartois, Kunzweiler, Maino, Pope):
  - In dimension 1, a 2^{602} isogeny over a field of 2360 bits: decomposition in 0.27s, image in 0.008s.
  - In dimension 2: decomposition in 0.49s, image in 0.025s (theta)
  - Richelot: decomposition in 4.85s, image in 0.47s

- Implementation in dimension 4 (Dartois): A 2^{128}-isogeny over a field of 500 bits in 0.62s.

Conclusion

Meme: funeral

- SIDH
- 2011-2022

Polarisations and isogenies on an abelian variety

- Polarisation on A = a (symmetric) isogeny $\lambda_A : A \to \widehat{A}$
- Principal polarisation: λ_A is an isomorphism.
- Warning: A may have several non equivalent principal polarisations if $g > 1$.

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 $(\operatorname{Jac} C, \Theta_C)$ where C is an hyperelliptic curve of genus 2.

Polarisations and isogenies on an abelian variety

- Polarisation on A = a (symmetric) isogeny $\lambda_A : A \to \widehat{A}$
- Principal polarisation: λ_A is an isomorphism.
- Warning: A may have several non equivalent principal polarisations if $g > 1$.

- $f : (A, \lambda_A) \to (B, \lambda_B)$ N-isogeny between ppav: $f^* \lambda_B = N \lambda_A$.

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

- Dual isogeny: $\hat{f} : \hat{B} \to \widehat{A}$
- Contragredient isogeny: $\tilde{f} = \lambda_A^{-1} \hat{f} \lambda_B : B \to A$
- f N-isogeny $\Leftrightarrow \tilde{f} f = N \Leftrightarrow f \tilde{f} = N$.

- $\mathrm{Ker} f = \mathrm{Im} \left(\tilde{f} \mid B[N] \right)$.