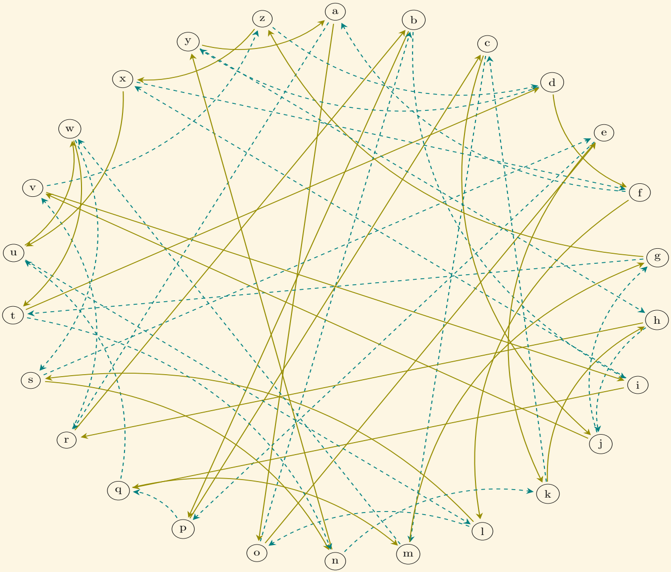
Efficient algorithms for abelian varieties and their moduli spaces

2021/06/15 — **HDR Defense** — Bordeaux

Damien Robert

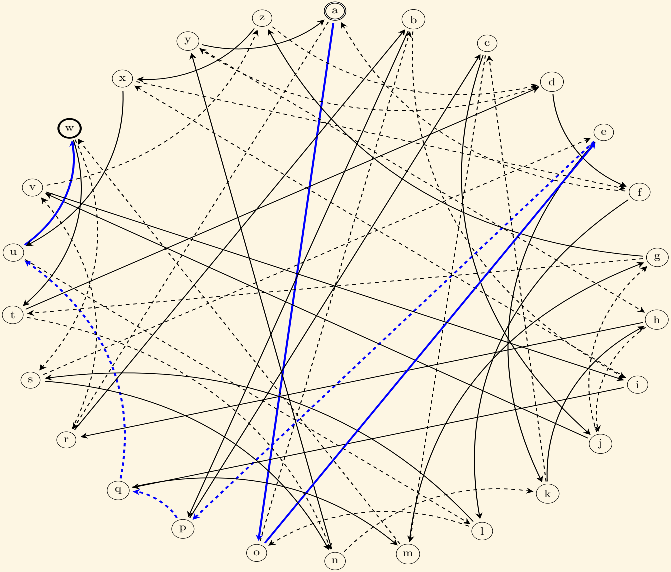
Outline

Key exchange by walking in graphs



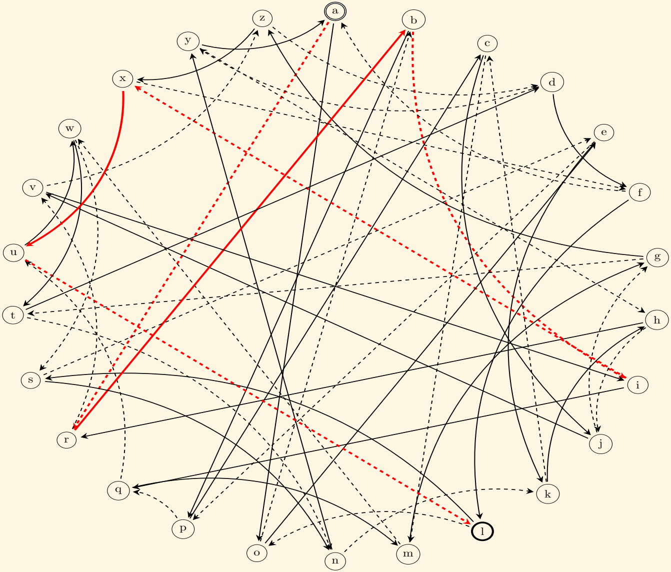
Key exchange by walking in graphs

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



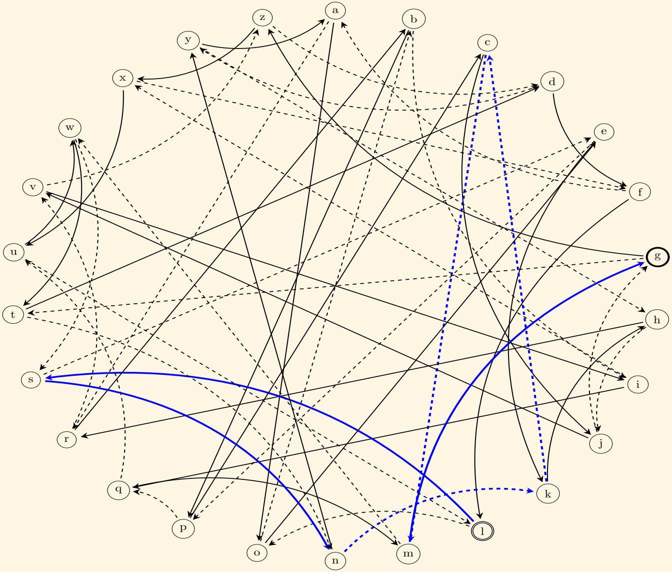
Key exchange by walking in graphs

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



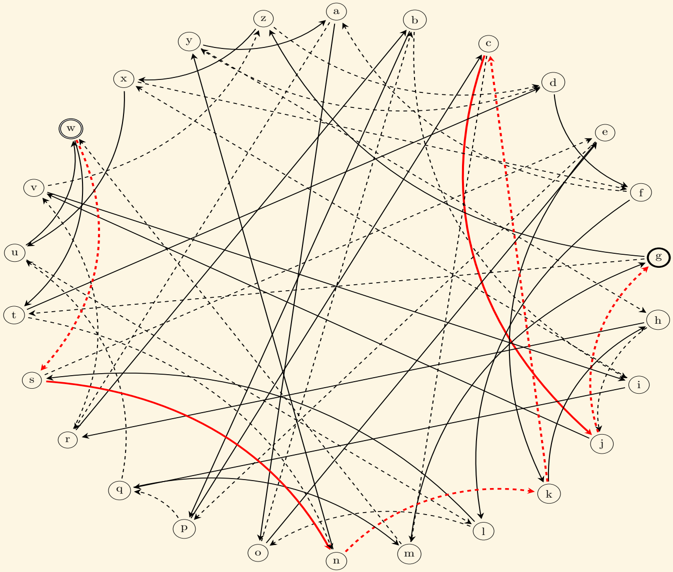
Key exchange by walking in graphs

Alice starts from ‘l’, follow her path 001110, and get ‘g’.



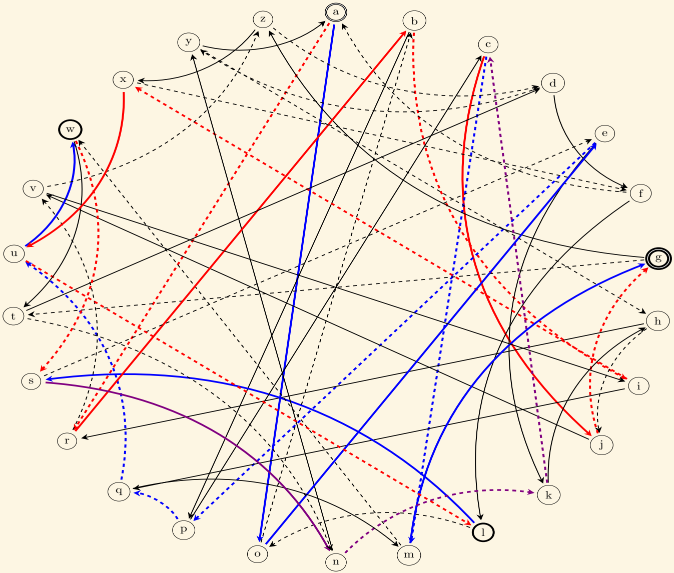
Key exchange by walking in graphs

Bob starts from ‘w’, follow his path 101101, and get ‘g’.



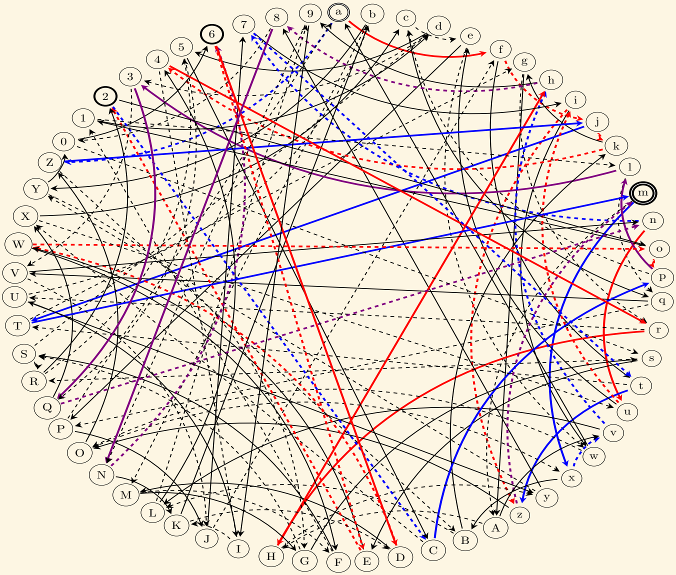
Key exchange by walking in graphs

The full key exchange



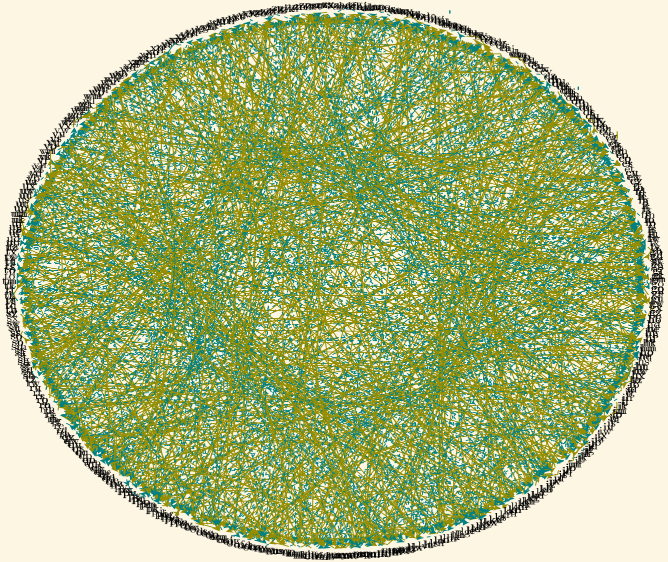
Key exchange by walking in graphs

Bigger graph (62 nodes)



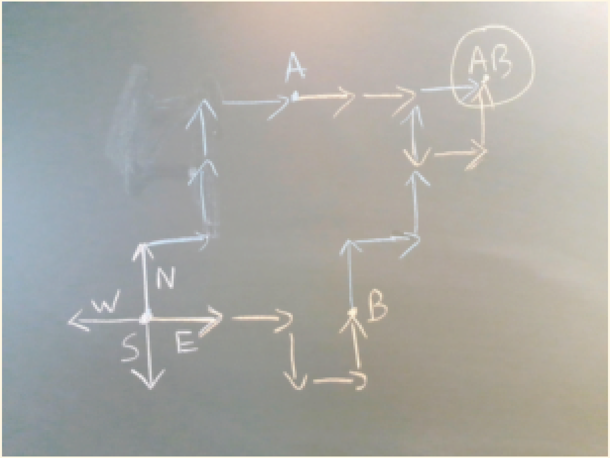
Key exchange by walking in graphs

Even bigger graph (676 nodes)



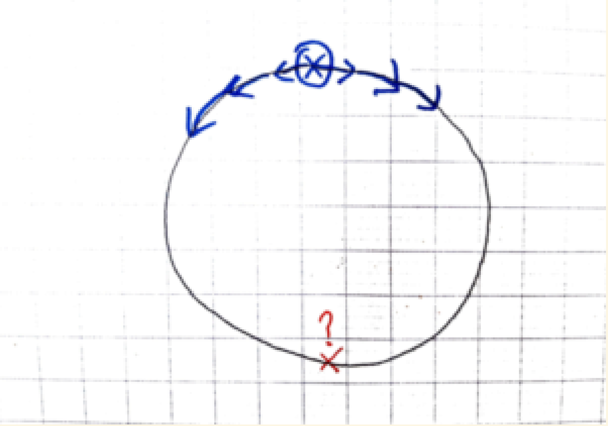
Security with N nodes

- Bad graphs:



Security with N nodes

- Bad graphs:



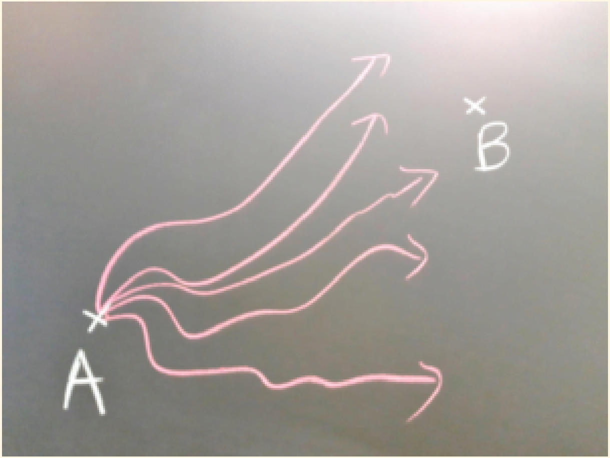
Security with N nodes

- Walking a short $m = O(\log N)$ path should give a random node: Ramanujan expander graphs.
  Examples: random graphs.
- Attack: find a path of length m between two nodes: $O(N)$.
- Meet in the middle: $O(\sqrt{N})$.
- Quantum (Grover): $O(N^{1/4})$.

$\Rightarrow$ 128 bits of security needs 2^{512} nodes.

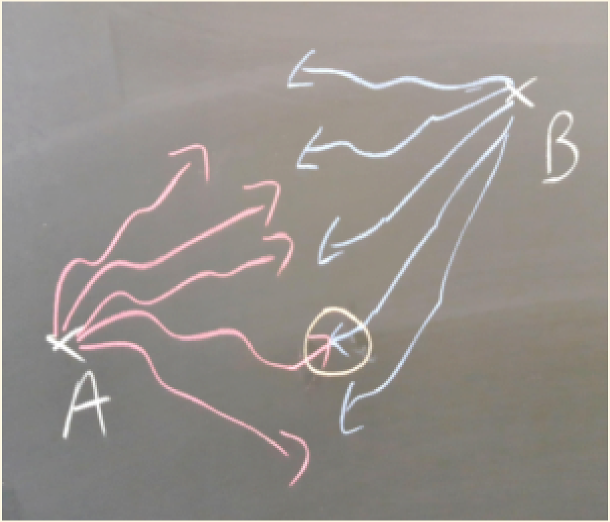
Security with N nodes

- Attack: find a path of length m between two nodes: $O(N)$.



Security with N nodes

- Meet in the middle: $O(\sqrt{N})$.



Security with N nodes

- Walking a short $m = O(\log N)$ path should give a random node: Ramanujan expander graphs.
  Examples: random graphs.
- Attack: find a path of length m between two nodes: $O(N)$.
- Meet in the middle: $O(\sqrt{N})$.
- Quantum (Grover): $O(N^{1/4})$.

$\Rightarrow$ 128 bits of security needs 2^{512} nodes.

Graph examples

- The Cayley graph of an abelian group G;
- The Schreier action graph of G acting on X;
- $\mathbb{Z}$ acts on G by $n \cdot g = g^n$. Walking on the graph $\approx$ fast exponentiation.
  Diffie-Hellman key exchange [DH76]: finding a path = DLP (discrete logarithm problem).
- $G = (E, +, 0_E)$ the group law of an elliptic curve $E/\mathbb{F}_q$ (ECC).
  DLP: exponential (classical) / polynomial (quantum: Schorr's algorithm).
  Classical cryptosystem.
- $E/\mathbb{F}_q$ ordinary elliptic curve with CM by O_K, $G = \mathrm{Cl}(O_K)$ acts on $X = \{$CM curves isogenous to $E\}$.
  Security if G abelian but not cyclic: exponential (classical) / subexponential (quantum);
- Graph of supersingular elliptic curves over $\mathbb{F}_{p^2}$: exponential security (classical and quantum).
  Post quantum cryptosystem.
- Non commutative graph: key exchange needs extra informations.

Polarised abelian varieties over $\mathbb{C}$

Definition

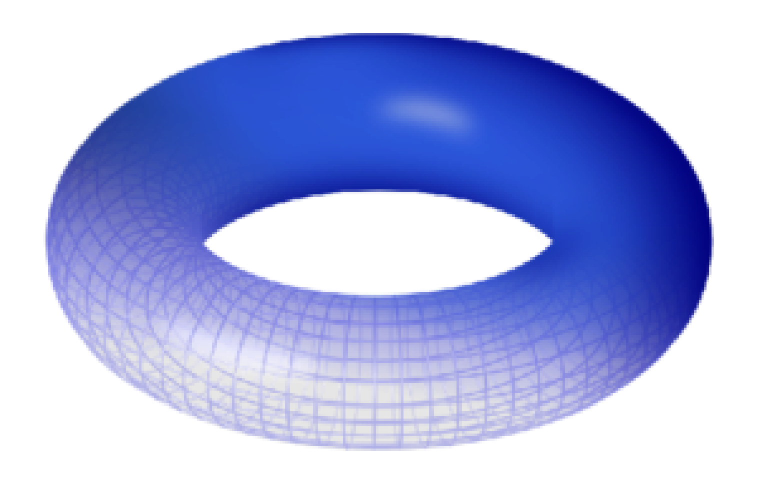
Complex abelian variety of dimension g: $A = V/\Lambda$,

- V: complex vector space of dimension g (linear data);
- Λ: $\mathbb{Z}$-lattice of rank $2g$ (arithmetic data);
- H: Hermitian form on V such that $E(\Lambda, \Lambda) \subset \mathbb{Z}$ where $E = \operatorname{Im} H$ is symplectic (quadratic data: pairings).

- H: polarisation ($\simeq$ algebraic class of an ample line bundle / divisor);
- Degree of H = degree of the kernel $\Lambda^\perp/\Lambda$ of the symplectic form E;
- H principal $\Leftrightarrow \deg H = 1$.

Polarised abelian varieties over $\mathbb{C}$

Dimension 1



Principal polarisations

- $A = \mathbb{C}^g/(\tau\mathbb{Z}^g \oplus \mathbb{Z}^g)$, $V = \mathbb{C}^g$, $\Lambda = \tau\mathbb{Z}^g \oplus \mathbb{Z}^g$;
- $\tau \in \mathfrak{H}_g$, the Siegel space of symmetric matrices τ with $\operatorname{Im}\tau$ positive definite;
- $H = (\operatorname{Im}\tau)^{-1}$, $E(\tau x_1 + x_2, \tau y_1 + y_2) = x_1 \cdot y_2 - x_2 \cdot y_1$.
- Moduli space of principally polarised abelian varieties: $\mathcal{A}_g = \mathfrak{H}_g / \operatorname{Sp}_{2g}(\mathbb{Z})$, where

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \tau = (a\tau + b)(c\tau + d)^{-1};$$

- Dimension: $g(g+1)/2$.

Coordinates

- Coordinates on (A, H): $f(x+\lambda) = a_H(\lambda, x) f(x) \quad \forall x \in V, \lambda \in \Lambda$,

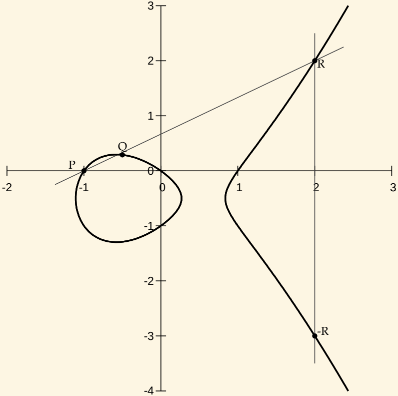
$$a_H(\lambda, x) = \pm e^{\pi\left(H(x,\lambda) + \frac{1}{2}H(\lambda,\lambda)\right)}$$

- $A = \mathbb{C}^g / (\tau\mathbb{Z}^g + \mathbb{Z}^g)$, $H_1 := (\operatorname{Im}\tau)^{-1}$ principal,
  $H := \ell H_1$ polarisation of level ℓ,
  Coordinates automorphic for H = vector space of dimension ℓ^g.
- Basis given by theta functions:

$$\theta\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right](z, \tau) = \sum_{n \in \mathbb{Z}^g} e^{\pi i\, {}^t(n+a)\tau(n+a) + 2\pi i\, {}^t(n+a)(z+b)} \qquad a, b \in \mathbb{Q}^g.$$

Dimension 1: elliptic curves

$E : y^2 = x^3 + ax + b. \quad \Delta := -16(4a^3 + 27b^2) \neq 0.$



$$P + Q = -R = (x_R, -y_R)$$
$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$$
$$x_R = \lambda^2 - x_P - x_Q$$
$$y_R = y_P + \lambda(x_R - x_P)$$

Dimension 1: elliptic curves

$E : y^2 = x^3 + ax + b. \quad \Delta := -16(4a^3 + 27b^2) \neq 0.$

- x, y: Weierstrass coordinates on E.
- a, b "coordinates" on the moduli space $\mathcal{A}_1$,
- Isomorphisms: $(x, y) \mapsto (X = u^2 x, Y = u^3 y)$

$$E : y^2 = x^3 + ax + b \to E' : Y^2 = X^3 + au^4 X + bu^6.$$

- Modular invariant: $j : \mathcal{A}_1 \to \mathbb{P}^1 \quad (\overline{\mathcal{A}_1} \simeq \mathbb{P}^1)$,

$$j(E) = 1728 \frac{4a^3}{4a^3 + 27b^2},$$

- Moduli space of dimension 1.

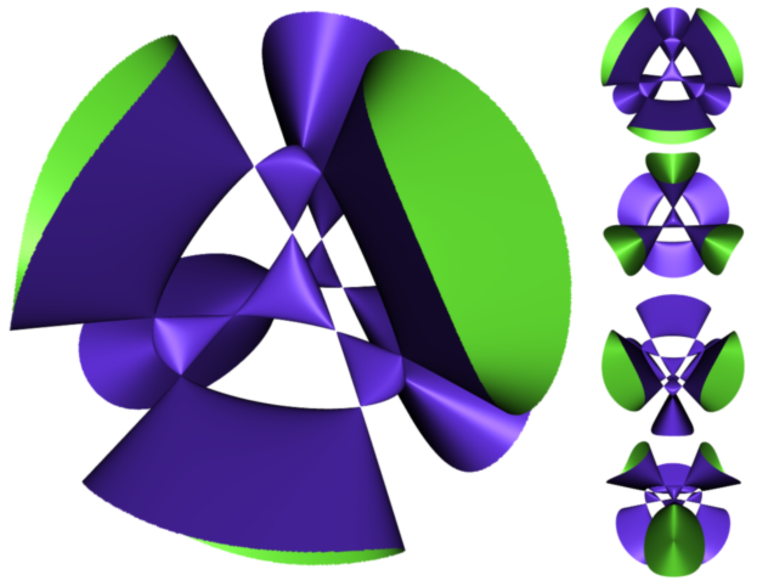
Dimension 2: abelian surfaces

- Kummer surfaces:

$$A(x^4+y^4+z^4+t^4)+Bxyzt+C(x^2y^2+z^2t^2)+D(x^2t^2+y^2z^2)+E(x^2z^2+y^2t^2)=0, \quad \Delta=0;$$

- Moduli space of dimension 3, birational to $\mathbb{P}^3$;
- Three Igusa invariants j_1, j_2, j_3.

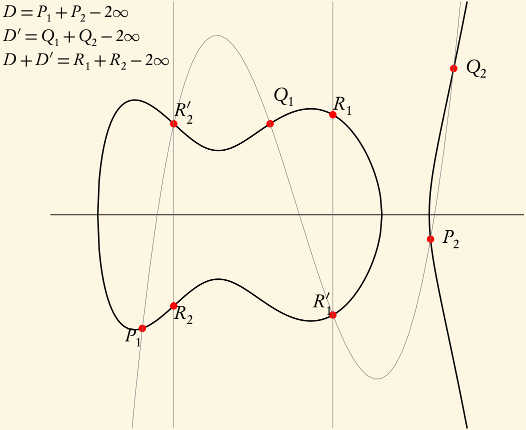
Dimension 2: abelian surfaces



Credit: Wikimedia.

Dimension 2: Jacobians of hyperelliptic curves of genus 2

$C/k : y^2 = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$



Coordinates on $\mathrm{Jac}(C)$: $x(P) + x(Q)$, $x(P)x(Q)$, $y(P)y(Q)$, $\frac{y(Q)-y(P)}{x(Q)-x(P)}$.

Dimension 2: Jacobians of hyperelliptic curves of genus 2

$C/k : y^2 = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$

- Up to isomorphism (over $\bar{k}$),
$$C : y^2 = x(x-1)(x-\lambda)(x-\mu)(x-\nu);$$
- Modular invariants: $\lambda + \mu + \nu, \lambda\mu + \lambda\nu + \mu\nu, \lambda\mu\nu.$

Isogenies

- Isogeny $\phi : A = V_1/\Lambda_1 \to B = V_2/\Lambda_2$ = bijective linear map $\phi : V_1 \to V_2$ with $\phi(\Lambda_1) \subset \Lambda_2$;
- Kernel: $\phi^{-1}(\Lambda_2)/\Lambda_1 \subset A$ is finite;
- Degree $\deg \phi$: cardinal of the kernel.

$\Rightarrow$ Isogeny graphs.

Algorithmic aspects of isogeny graphs

1. Given A, K, compute $B = A/K$ and the isogeny $\phi : A \to B$ (follow a direction).
2. Given A, list all isogenous B (find neighboors).
3. Given isogenous A, B, find $\phi : A \to B$ or $K = \operatorname{Ker} \phi$ (find a path).

Isogenies and polarisations

- Given A, K, compute $B = A/K$ and the isogeny $\phi : A \to B$.

Isogenies and polarisations

- Given
  1. coordinates $f_1, \dots, f_m$ on A automorphic for H_A (of level n),
  2. a kernel K expressed in these coordinates,

  construct
  1. $B = A/K$,
  2. a polarisation H_B (of level n),
  3. coordinates $g_1, \dots, g_m$ on B automorphic for H_B,

  and express $g_i \circ \phi$ in terms of the f_i.

Isogenies and polarisations

- If g_i is automorphic for H_B, $g_i \circ \phi$ is automorphic for
$$\phi^* H_B := H_B(\phi(\cdot), \phi(\cdot));$$
- H'_A is of the form $\phi^* H_B$ iff
$$\operatorname{Im} H'_A(K + \Lambda_A, K + \Lambda_A) \subset \mathbb{Z}$$
iff K is isotropic for the E'_A-pairing.
- ϕ ℓ-isogeny: $H'_A := \phi^* H_B = \ell H_A$
$\Leftrightarrow K$ maximal isotropic for the Weil pairing e_{ℓ, H_A}.

- If f is automorphic for H'_A, it is of the form $f = g \circ \phi$ iff f is invariant by translation by K.

- Step 1: from the coordinates $f_1, \ldots, f_m$ construct coordinates automorphic for ℓH_A;
- Step 2: find coordinates invariant by translation (eg taking a trace).

Vélu's formula for ℓ-isogeny for elliptic curves [Vél71]

- Weierstrass coordinates x, y on E are sections of the divisor $3(0_E)$.
- $D := \sum_{T \in K} 3(T) \sim 3\ell(0_E)$ descends to E/K;
- $X(P) = \sum_{T \in K \backslash 0_E} x(P+T)$, $Y(P) = \sum_{T \in K \backslash 0_E} y(P+T)$ descend to E/K: Weierstrass coordinates of E/K.

Vélu's formula for ℓ-isogeny in higher dimension

- In dimension g, if Θ_A is principal on A, $3\Theta_A$ is very ample of level $n = 3$ and

$$\sum_{T \in K} \tau_T^* 3\Theta_A \sim 3\ell^g \Theta_A$$

descends to $B = A/K$;

- If f is a section of $3\Theta_A$, $F(P) = \sum_{T \in K} f(P+T)$ is invariant by K so descends to B;
- But F is automorphic for $3\ell^g H_{\Theta_A}$ on A of level $3\ell^g$, so descends to a function of level $3\ell^{g-1}$ on B.

- $3\ell\Theta_A$ does not descend to B;
- But it is isomorphic to a divisor which descends to a divisor on B of level 3.
- Theta group $G(3\ell\Theta_A)$: encodes the isomorphisms of $3\ell\Theta_A$.
- Descending $3\ell\Theta_A \Leftrightarrow$ level subgroup above K (Grothendieck's fpqc descent theory).

- Quasi-linear algorithm: [Cosset, Dudeanu, Jetchev, Lubicz, R., Vuille].

Modular polynomials for elliptic curves

Definition (Modular polynomial)

The modular polynomial $\phi_\ell(x,y) \in \mathbb{Z}[x,y]$ is a bivariate polynomial such that $\phi_\ell(x,y) = 0 \Leftrightarrow x = j(E_1)$ and $y = j(E_2)$ with E_1 and E_2 ℓ-isogeneous.

- Roots of $\phi_\ell(j(E_1),\cdot) \Leftrightarrow$ elliptic curves ℓ-isogeneous to E_1.
  There are $\ell+1 = \#\mathbb{P}^1(\mathbb{F}_\ell)$ such roots if ℓ is prime.
- ϕ_ℓ is symmetric (dual isogenies);
- Degree $\ell+1$ in x and y;
- Height: $\widetilde{O}(\ell)$

$\Rightarrow$ Total size: $\widetilde{O}(\ell^3)$.

Example

$\phi_3(x,y) = x^4 + y^4 - x^3y^3 + 2232x^3y^2 + 2232x^2y^3 - 1069956x^3y - 1069956xy^3 + 36864000x^3 + 36864000y^3 + 2587918086x^2y^2 + 8900222976000x^2y + 8900222976000xy^2 + 452984832000000x^2 + 452984832000000y^2 - 770845966336000000xy + 1855425871872000000000x + 1855425871872000000000y.$

Modular polynomials for abelian surfaces [Milio]

Definition (Siegel modular polynomials)

The modular polynomials $\Phi_\ell(X,Y) \in \mathbb{Q}(X)[Y]$ parametrize Igusa j-invariants $X = (j_1(A), j_2(A), j_3(A))$ and $Y = (j_1(B), j_2(B), j_3(B))$ of ℓ-isogenous abelian surfaces.

- Computed via a multidimensional evaluation–interpolation approach.
- Requires evaluating modular invariants on τ and period matrices from invariants at high precision;

$\Rightarrow$ generalized version of the AGM to compute theta functions in quasi-linear time in the precision [Dupont: Dup06];

$\Rightarrow$ Need to interpolate rational functions;

- Denominator describes the Humbert surface of discriminant ℓ^2 [Bröker, Gruenewald, Lauter: BL09; Gru10]: abelian surfaces ℓ-isogenous to product of elliptic curves;
- Quasi-linear algorithm [Dup06; Mil15];
- Generalized to smaller modular invariants [Milio: Mil15].
- Hilbert modular polynomials [Milio-R.: MR20] for β-isogenies, $\beta \in \mathrm{End}^+(A)$ (+ modular interpretation of their denominators).

Example of modular polynomials in dimension 2 [Milio: Mil15]

Invariant	ℓ	Size
Igusa	2	57 MB
Streng	2	2.1 MB
Streng	3	890 MB
Theta	3	270 KB
Theta	5	305 MB
Theta	7	29 GB

Examples (Theta invariants)

- Denominator of Φ_3:
  $1024 b_3^6 b_2^6 b_1^{10} - ((768 b_3^8 + 1536 b_3^4 - 256) b_3^8 + 1536 b_3^8 b_3^4 - 256 b_3^8) b_1^8 + (1024 b_3^6 b_2^{10} + (1024 b_3^{10} + 2560 b_3^6 - 512 b_3^2) b_2^6 - (512 b_3^6 - 64 b_3^2) b_2^2) b_1^6 - (1536 b_3^8 b_2^8 + (-416 b_3^4 + 32) b_2^4 + 32 b_3^4) b_1^4 - ((512 b_3^6 - 64 b_3^2) b_2^6 - 64 b_3^6 b_2^2) b_1^2 + 256 b_3^8 b_2^8 - 32 b_3^4 b_2^4 + 1.$
- One coefficient of the denominator for Φ_5 is 1180591620717411303424.

Example of cyclic modular polynomials in dimension 2 [Milio-R.: MR20]

ℓ ($\mathbb{Q}(\sqrt{2})$)	Size (Gundlach)	Theta	ℓ ($\mathbb{Q}(\sqrt{5})$)	Size (Gundlach)	Theta
2	8.5 KB		5	22 KB	45 KB
7	172 KB		11	3.5 MB	308 KB
17	5.8 MB	221KB	19	33 MB	3.6 MB
23	21 MB		29	188 MB	21 MB
31	70 MB		31	248 MB	28 MB
41	225 MB	7.2 MB	41	785 MB	115 MB
73		81 MB	59	3600 MB	470 MB
89		188 MB			
97		269 MB			

Examples (Pullback of theta invariants)

- For $D = 2$, $\beta = 5 + 2\sqrt{2} \mid 17$, the denominator of $\Phi_{1,\beta}$ is $b_3^6 b_2^{18} + (6b_3^8 - 6b_3^4 + 1)b_2^{16} + (15b_3^{10} - 24b_3^6 + 7b_3^2)b_2^{14} + (20b_3^{12} - 42b_3^8 + 9b_3^4 + 2)b_2^{12} + (15b_3^{14} - 48b_3^{10} + 37b_3^6 + 4b_3^2)b_2^{10} + (6b_3^{16} - 42b_3^{12} + 68b_3^8 - 26b_3^4 + 3)b_2^8 + (b_3^{18} - 24b_3^{14} + 37b_3^{10} + 8b_3^6 - b_3^2)b_2^6 + (-6b_3^{16} + 9b_3^{12} - 26b_3^8 - 24b_3^4 + 2)b_2^4 + (7b_3^{14} + 4b_3^{10} - b_3^6)b_2^2 + (b_3^{16} + 2b_3^{12} + 3b_3^8 + 2b_3^4 + 1)$.
- For $\beta \mid 97$, one coefficient of the denominator of $\Phi_{1,\beta}$ is $50853993476624629 2$.

Size of modular polynomials [Kieffer: Kie22]

- If the moduli space is of dimension N and the degree of the modular correspondance is D, the modular polynomials are
  - of total degree $O(D)$ in X and Y,
  - with coefficients of height $\widetilde{O}(D)$ [Kie22].
- Total size: $O(DD^N)$ terms of height $\widetilde{O}(D)$: $\widetilde{O}(D^{N+2})$.

- Siegel ℓ-modular polynomial:
  $N = g(g+1)/2$, $D = O(\ell^N)$, total size: $\widetilde{O}(\ell^{N(N+2)})$.
  Ex: $\widetilde{O}(\ell^3)$ for $g = 1$, $\widetilde{O}(\ell^{15})$ for $g = 2$, $\widetilde{O}(\ell^{48})$ for $g = 3$.

- Hilbert β-modular polynomial:
  $N = g$, $D = O(\ell)$, $\ell := O(N(\beta))$, total size $\widetilde{O}(\ell^{g+2})$.
  Ex: $\widetilde{O}(\ell^4)$ for $g = 2$.

Evaluating modular polynomials over $\mathbb{F}_p$ [Kieffer: Kie20], [R.]

- Goal: for $A/\mathbb{F}_p$, evaluate $\Phi_\ell(J(A), Y)$;
- Strategy: lift A to $\widetilde{A}/\mathbb{Q}$, evaluate over $\mathbb{Q}$ and reduce modulo p;
- If $J(\widetilde{A})$ is of height $H = O(\log p)$, $\Phi_\ell(J(\widetilde{A}), Y)$ has $O(D)$ coefficients of height $\widetilde{O}(DH)$, total size: $\widetilde{O}(D^2H)$.

- Analytic method in dimension 1 (folklore?):

$$\widetilde{O}(D^2H) = \widetilde{O}(\ell^2 \log p).$$

  Via explicit CRT [Sutherland: Sut13]: $\widetilde{O}(\ell^3 + \ell^2 \log p)$.
- Analytic method in dimension 2 [Kieffer: Kie20]:

$$\widetilde{O}(D^2H + DH^2).$$

  Ex: $\widetilde{O}(\ell^2 \log p + \ell \log^2 p)$ for Hilbert.
  (Dimension $g > 2$ lacks fast period matrix from invariants).
- [R.]: p-adic and CRT method in any dimension in $\widetilde{O}(ED^2H)$,
  E = cost of evaluating one isogeny (Siegel: $E = \widetilde{O}(\ell^g)$, Hilbert: $E = \widetilde{O}(\ell)$).

☹ None are quasi-linear over $\mathbb{Q}$ (except analytic when $g = 1$).
See my hdr for possible strategies.

Recovering an isogeny

- Goal: given ℓ-isogenous $E_1 : y^2 = x^3 + ax + b$, $E_2 : Y^2 = X^3 + AX + B$, recover the isogeny $\phi : E_1 \to E_2$ or the kernel $K = \operatorname{Ker} \phi$;
- $w_E = dx/y$, $\phi^* w_{E_2} = M w_{E_1}$.
  $M = 1$: ϕ normalised isogeny;
- $\phi(x,y) = (h(x), \frac{1}{M} y h'(x))$, $h(x) \in k(x)$, $\quad dX/Y = M \frac{h'(x) dx}{y h'(x)} = M dx/y$.
- Differential equation:

$$\frac{1}{M^2}(x^3 + ax + b) h'(x)^2 = h(x)^3 + A h(x) + B.$$

- Newton iterations + rational reconstruction in $k[[x]]$: $h(x)$ in quasi-linear time ($p \gg \ell$).
- Problem: need M.

Recovering an isogeny between elliptic curves [Elkies: Elk97]

- An isomorphism $E_2 \simeq E_2'$, $(X, Y) \mapsto (u^2 X, u^3 Y)$ maps $w_{E_2} = dX/Y$ to $\frac{1}{u} w'_{E_2} = \frac{1}{u} dX/Y$, so changes M by a factor u.
- Need: a covariant g that depends on E and w_E: $g(E, uw_E) = u^{-k} g(E, w_E)$.
- Modular function of weight k (+ boundary conditions).
- Period matrices: $E : \mathbb{C}/(\mathbb{Z} + \tau\mathbb{Z})$, $w_E = 2\pi i dz$,

$$g(\frac{a\tau + b}{c\tau + d}) = (c\tau + d)^k g(\tau).$$

- $a(\tau)$ modular form of weight 4, $b(\tau)$ modular form of weight 6;
- $j'(\tau) = 18 j(\tau) \frac{b(\tau)}{a(\tau)}$ modular function of weight 2;
- Algebraic interpretation: $j(E_\epsilon) = j(E) + j'(E, w_E)\epsilon$, E_ϵ the deformation corresponding to $w_E^{\otimes 2}$ via the Kodaira-Spencer isomorphism;
- Differentiating $\Phi_\ell(j(E_1), j(E_2)) = 0$:

$$j'(E_1, w_{E_1}) \frac{\partial \Phi}{\partial x}(j(E_1), j(E_2)) + j'(E_2, w_{E_2}) \frac{\partial \Phi}{\partial y}(j(E_1), j(E_2)).$$

Encodes an $M = \sqrt{\ell}$-normalised isogeny.

Recovering an isogeny between abelian surfaces [Kieffer-Page-R.]

- Goal: given ℓ-isogenous Jacobians of the curves $C_1 : y^2 = h_1(x)$, $C_2 : Y^2 = h_2(X)$, recover the isogeny ϕ.
- $w_{\mathrm{Jac}(C)} = (xdx/y, dx/y)$, $\phi^* w_{\mathrm{Jac}(C_2)} = M w_{\mathrm{Jac}(C_1)}$, $M = \begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix}$ a 2×2 matrix;
- Differential equation:

$$\begin{cases} \dfrac{X_1\,dX_1}{Y_1} + \dfrac{X_2\,dX_2}{Y_2} & = & (m_{1,1}x + m_{1,2})\dfrac{dx}{y} \\ \dfrac{dX_1}{Y_1} + \dfrac{dX_2}{Y_2} & = & (m_{2,1}x + m_{2,2})\dfrac{dx}{y} \\ Y_1^2 = h_2(X_1) \\ Y_2^2 = h_2(X_2), \end{cases}$$

- Newton iterations + rational reconstruction: ϕ in quasi-linear time ($p \gg \ell$).

- If $J(\tau) = (j_1(\tau), j_2(\tau), j_3(\tau))$, $J'(\tau)$ is a vectorial modular function of weight Sym^2 (Kodaira-Spencer isomorphism);
- Differentiating $\Phi_\ell(J(A_1), J(A_2))$ recovers the 3×3 matrix $\mathrm{Sym}^2(M)$.
- Formula for $J'(\mathrm{Jac}\, C, w_{\mathrm{Jac}\, C})$ in terms of the coefficients of C [KPR25].

⇒ Application to fast point counting for $g = 2$ [Kieffer]: $\widetilde{O}(\log^4 p)$ SEA-like algorithm in the Hilbert case.

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation $\Rightarrow$ fast decompression $(p \gg \ell)$.
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation $\Rightarrow$ fast decompression $(p \gg \ell)$.
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation ⇒ fast decompression $(p \gg \ell)$.
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation $\Rightarrow$ fast decompression $(p \gg \ell)$.
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation $\Rightarrow$ fast decompression ($p \gg \ell$).
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Compressing an isogeny [R.]

- ϕ is determined by $K \subset E/\mathbb{F}_p$: size $O(\ell \log p)$;
- If $K = \langle T \rangle$, $T \in E(\mathbb{F}_p)$, ϕ is determined by T: size $O(\log p)$.
- General case: encode ϕ via $(T, \phi(T))$, $T \in E(\mathbb{F}_p)$ of order $N \gg \ell$. Size: $O(\log p)$.
- Better idea: take T a fat $k[\epsilon]$-point over 0_E. It is of order p.
  Encodes M, ie the differential equation ⇒ fast decompression $(p \gg \ell)$.
- Lift if p is too small.

Proposition (Slogan)

- *The isogeny ϕ is efficiently encoded by the normalised (lifted) $j(E_1), j'(E_1), j(E_2), j'(E_2)$: size $O(\log \ell + \log p)$.*
- *All the ℓ-isogenies from E_1 are efficiently encoded by the evaluated modular polynomials:*

$$\Phi_\ell(j(E_1), y), \quad \partial\Phi_\ell/\partial_x(j(E_1), y).$$

  Computed in time $\widetilde{O}(\ell^2 \log p)$.
- *Rational roots of $\Phi_\ell(j(E_1), y)$: $\widetilde{O}(\ell \log^2 p)$; decompression of a kernel: $\widetilde{O}(\ell \log p)$.*
- *Improves the complexity $\widetilde{O}(\ell^2 \log^6 p)$ for one isogeny of [De Feo, Hugounenq, Plût, Schost: DHPS16] (Couveignes' method).*

Point counting in small characteristic

- $E/\mathbb{F}_q$ ordinary elliptic curve, π_q Frobenius, $\pi_q^* w_E = \lambda_q w_E$, $t = \lambda_q + q/\lambda_q$,

$$\#E(\mathbb{F}_q) = q + 1 - t.$$

- Problem: π_q is of degree q;
- Solution: if $q = p^d$, $\pi_p^* w_E^{\pi_p} = \lambda_p w_E$, then $\lambda_q = N_{\mathbb{F}_q/\mathbb{F}_p}(\lambda_p)$;
- π_p is easy to compute if p is small;

- Problem: only get $\lambda_p \mod p$; not enough information.
- Solution (Satoh [Sat00]): lift to $\mathbb{Q}_q$.

Satoh's algorithm ([Maiga-R.] for $g = 2$)

1. Compute the canonical lift $\widetilde{E}/\mathbb{Q}_q$.
2. Lift the kernel of the Frobenius/Verschiebung;
3. Compute the isogeny over $\mathbb{Q}_q$;
4. Recover $\lambda_p \in \mathbb{Q}_q$ with enough p-adic precision m $\quad (m = O(d))$;
5. Take the norm and recover $t \in \mathbb{Z}$.

Optimal complexity: $\widetilde{O}(dm) = \widetilde{O}(d^2)$.

Satoh's algorithm ([Maiga-R.] for $g = 2$)

1. Compute the canonical lift $\widetilde{E}/\mathbb{Q}_q$.
   Solve $\phi_p(j(\widetilde{E}), \sigma_p(j(\widetilde{E}))) = 0$ via Newton iterations.
2. Lift the kernel of the Frobenius/Verschiebung;
3. Compute the isogeny over $\mathbb{Q}_q$;
4. Recover $\lambda_p \in \mathbb{Q}_q$ with enough p-adic precision m $(m = O(d))$;
5. Take the norm and recover $t \in \mathbb{Z}$.

Optimal complexity: $\widetilde{O}(dm) = \widetilde{O}(d^2)$.

Satoh's algorithm ([Maiga-R.] for $g = 2$)

1. Compute the canonical lift $\widetilde{E}/\mathbb{Q}_q$.
   Solve $\phi_p(j(\widetilde{E}), \sigma_p(j(\widetilde{E}))) = 0$ via Newton iterations.
2. ~~Lift the kernel of the Frobenius/Verschiebung~~;
3. ~~Compute the isogeny over $\mathbb{Q}_q$~~;
4. Recover $\lambda_p \in \mathbb{Q}_q$ with enough p-adic precision m $(m = O(d))$;
5. Take the norm and recover $t \in \mathbb{Z}$.

Optimal complexity: $\widetilde{O}(dm) = \widetilde{O}(d^2)$.

Improved version of Satoh's algorithm [R.]

q	Time (old)	Memory (old)	Time (new)	Memory (new)
11^{1008}	48.5s	512MB	4.5s	128MB
101^{102}	91s	1024MB	9s	128MB
101^{256}	633s	4096MB	26s	128MB
101^{310}	924s	8192MB	35s	256MB
101^{418}	1813s	16384MB	55s	256MB

Bibliography

[BL09] R. Bröker and K. Lauter. "Modular polynomials for genus 2". In: LMS J. Comput. Math. 12 (2009), pp. 326–339. issn: 1461-1570. arXiv: 0804.1565 (cit. on p. 36).

[CR15] R. Cosset and D. Robert. "An algorithm for computing (ℓ, ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". In: Mathematics of Computation 84.294 (Nov. 2015), pp. 1953–1975. doi: 10.1090/S0025-5718-2014-02899-8. url: http://www.normalesup.org/~robert/pro/publications/articles/niveau.pdf. eprint: 2011/143, HAL: hal-00578991.

[DHPS16] L. De Feo, C. Hugounenq, J. Plût, and É. Schost. "Explicit isogenies in quadratic time in any characteristic". In: LMS Journal of Computation and Mathematics 19.A (2016), pp. 267–282 (cit. on pp. 44–49).

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: IEEE Transactions on information Theory 22.6 (1976), pp. 644–654 (cit. on p. 17).

[DJRV22] A. Dudeanu, D. Jetchev, D. Robert, and M. Vuille. "Cyclic Isogenies for Abelian Varieties with Real Multiplication". In: Moscow Mathematical Journal 22 (Feb. 2022), pp. 613–655. url: http://www.normalesup.org/~robert/pro/publications/articles/cyclic.pdf. HAL: hal-01629829.

[Dup06] R. Dupont. "Moyenne arithmetico-geometrique, suites de Borchardt et applications". In: These de doctorat, Ecole polytechnique, Palaiseau (2006) (cit. on p. 36).

[Elk97] N. Elkies. "Elliptic and modular curves over finite fields and related computational issues". In: Computational perspectives on number theory: proceedings of a conference in honor of AOL Atkin, September 1995 Vol. 7. Amer Mathematical Society. 1997, p. 21 (cit. on p. 42).

[Gru10] D. Gruenewald. "Computing Humbert surfaces and applications". In: Arithmetic, Geometry, Cryptography and Codint Theory 2009 (2010), pp. 59–69 (cit. on p. 36).

[IMRS14] S. Ionica, C. Martindale, D. Robert, and M. Streng. "Isogeny graphs of ordinary abelian surfaces over a finite field". Mar. 2014. In preparation.

[Kie20] J. Kieffer. "Evaluating modular polynomials in genus 2". 2020. arXiv: 2010.10094 [math.NT]. HAL: hal-02971326. (Cit. on p. 40).

[Kie22] J. Kieffer. "Degree and height estimates for modular equations on PEL Shimura varieties". In: Journal of the London Mathematical Society 105.2 (2022), pp. 1314–1361 (cit. on p. 39).

[KPR25] J. Kieffer, A. Page, and D. Robert. "Computing isogenies from modular equations between Jacobians of genus 2 curves". In: Journal of Algebra 666 (Mar. 2025), pp. 331–386. doi: 10.1016/j.jalgebra.2024.11.029. arXiv: 2001.04137 [math.AG]. url: http://www.normalesup.org/~robert/pro/publications/articles/modular_isogenies_g2.pdf. HAL: hal-02436133. (Cit. on p. 43).

[LR12] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: Compositio Mathematica 148.5 (Sept. 2012), pp. 1483–1515. doi: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG]. url: http://www.normalesup.org/~robert/pro/publications/articles/isogenies.pdf. HAL: hal-00446062.

[LR15] D. Lubicz and D. Robert. "Computing separable isogenies in quasi-optimal time". In: LMS Journal of Computation and Mathematics 18 (1 Feb. 2015), pp. 198–216. doi: 10.1112/S146115701400045X. arXiv: 1402.3628. url: http://www.normalesup.org/~robert/pro/publications/articles/rational.pdf. HAL: hal-00954895.

[Mil15] E. Milio. "A quasi-linear time algorithm for computing modular polynomials in dimension 2". In: LMS Journal of Computation and Mathematics 18.1 (2015), pp. 603–632. arXiv: 1411.0409 (cit. on pp. 36, 37).

[MR20] E. Milio and D. Robert. "Modular polynomials on Hilbert surfaces". In: Journal of Number Theory 216 (Nov. 2020), pp. 403–459. doi: 10.1016/j.jnt.2020.04.014. url: https://www.sciencedirect.com/science/article/abs/pii/S0022314X20301402. HAL: hal-01520262, Reproducible archive: https://data.mendeley.com/datasets/yy3bty5ktk/1. (Cit. on pp. 36, 38).

[Sat00] T. Satoh. "The canonical lift of an ordinary elliptic curve over a finite field and its point counting". In: J. Ramanujan Math. Soc. 15.4 (2000), pp. 247–270 (cit. on p. 50).

[Sut13] A. Sutherland. "On the evaluation of modular polynomials". In: The Open Book Series 1.1 (2013), pp. 531–555 (cit. on p. 40).

[Vél71] J. Vélu. "Isogénies entre courbes elliptiques". In: Compte Rendu Académie Sciences Paris Série A-B 273 (1971), A238–A241 (cit. on p. 33).