PCMI 2022: Supersingular isogeny graphs in cryptography

Exercises Lecture 1: Elliptic curves, Isogenies, CGL Hash Function

TA: Jana Sotáková **version July 25, 2022**

You can find hints and more explanation for the first three exercises on the following page.
Please note that we can use the Zulip stream on Drew's server to ask questions!
We will try to keep the Whova and PCMI websites up to date but the quickest updates will be here.

1. (Elliptic curves) Over $\mathbb{F}_p$ for $p = 431$:
   (a) Define an elliptic curve $E/\mathbb{F}_p$ with $E : y^2 = x^3 + x$.
   (b) Compute its j-invariant;
   (c) Find an elliptic curve $E_1/\mathbb{F}_p$ with j-invariant 234;
   (d) Is this elliptic curve supersingular?
   (e) Find another elliptic curve E_2 with j-invariant 234 such that E_1 and E_2 are not isomorphic over $\mathbb{F}_p$. Find the smallest extension $\mathbb{F}_q/\mathbb{F}_p$ such that E_1 and E_2 are isomorphic.

2. (Isogenies) Compute the following for $E : y^2 = x^3 + x/\mathbb{F}_{431^2}$
   (a) Isogeny $\varphi : E \to E'$ with kernel generated by $(0, 0)$. What is the degree?
   (b) Compute the dual isogeny $\hat{\varphi} : E' \to E$;
   (c) Find all the isogenies of degree 2 from E.
   (d) Find all the cyclic isogenies of degree 16 from E.
   (e) Compute a cyclic isogeny of degree 16 as a sequence of 2-isogenies.

3. (Modular polynomial) Use the modular polynomial $\Phi_N(X, Y)$ to find isogenous curves:
   (a) Find all the 2-isogenies curves to $E : y^2 = x^3 + 26x + 279/\mathbb{F}_{431^2}$;
   (b) Find j-invariants of elliptic curves admitting a 8-isogeny from E.
   (c) Find all the self-loops in the ℓ-isogeny graph for $\ell \leq 11$.

4. (Supersingular isogeny graphs) Write code to generate the supersingular isogeny graph over $\mathbb{F}_{p^2}$, using the following steps. On input coprime primes p and ℓ;
   (a) Find one supersingular elliptic curve over $E_0/\mathbb{F}_{p^2}$, represented by the j-invariant;
   (b) Write a neighbor function that on input an elliptic curve E, finds all the neighbours of E in the SSIG $\mathcal{G}_\ell$: (the j-invariants of) all the supersingular elliptic curves ℓ-isogenous to E.
   (c) Using a breadth-first-search approach, generate the graph by starting from the curve found in Step (b) and the Neighbor function from Step (c).

   You can visualize the supersingular isogeny graph for instance in Sage. You can use the code in your Sage installation or on Cocalc.

5. (CGL Hash function, details here and here.) For a small prime p and any starting supersingular elliptic curve E, find a collision for the CGL hash fuction on the 2-isogeny SSIG. I.e., find two strings that hash to the same elliptic curve. This requires you to decide on the ordering of the edges in the SSIG. Find two isogenies to the same curve.

Hints, comments, definitions

1. The elliptic curves $E : y^2 = x^3 \pm x$ always have j-invariant 1728 and are supersingular if and only if $p \equiv 3 \bmod 4$. Similarly, the curve $E : y^2 = x^3 \pm 1$ always has j-invariant 0 and is supersingular if and only if $p \equiv 2 \pmod 3$. Keep these two examples in mind.

   As Kristin mentioned in the lecture, two curves with the same j-invariant are isomorphic over $\bar{F}_p$ but not necessarily over the same field. Notably, (for $p > 3$) there will always be two isomoprhism classes over $\mathbb{F}_q$, and those curves are quadratic twists of each other (in the case $j(E) = 1728$ then quartic twists).

   Magma commands: `EllipticCurve, jInvariant, EllipticCurveWithjInvariant, IsSupersingular, QuadraticTwists, IsIsomorphic`

2. For curves in the Weierstrass form, points with $x = 0$ are always of order 2. So the isogeny should be of degree 2. The dual isogeny $\hat{\varphi}$ is such that

$$\hat{\varphi} \circ \varphi = [\deg \varphi] \quad \text{on } E,$$

   the dual isogeny will therefore also be an isogeny of degree 2. In isogeny graphs, we typically identify an isogeny with its dual to get the undirected edges. Note that this requires choices for $j = 0$ and $j = 1728$ because of automorphisms.

   Every isogeny is determined by its kernel, so all the 2-isogenies correspond to all the (cyclic) subgroups of size 2. So to find all the 2-isogenies, we need to determine all the points of order 2.

   Cyclic isogenies of degree 16 correspond to cyclic subgroups of degree 16. You can find them all by first finding a basis of the 16-torsion, that is, finding two independent points of order 16, and then finding all the cyclic subgroups of size 16.

   Remember that isogenies are the well-behaved group quotients for elliptic curves: the isogeny with kernel $\langle P \rangle$ is sometimes written as $E \to E/\langle P \rangle$. Moreover, if P has order 16, then the 16-isogeny with kernel $\langle P \rangle$ can be decomposed using intermediate subgroups as

$$E \to E/\langle 8P \rangle \to E/\langle 4P \rangle \to E/\langle 2P \rangle \to E/\langle P \rangle.$$

   Magma commands: `E![0,0]` defines $(0, 0)$ as a point on E; `Order` to compute order of a point, `Random(E)` to get a random point on E; To compute isogeny: `IsogenyFromKernel` requires a kernel polynomial; for point P of order 2, can get as follows:

```
R<X> := PolynomialRing(F); IsogenyFromKernel(E, X - P[1]);
```

   Easy way how to generate isogenies from points for any N:

```
function IsogenyFromPoint(P)  // Needs to have the poly ring in X defined!
  Edom := Curve(P);
  n := Order(P);
  return IsogenyFromKernel(Edom, &*[X - (i*P)[1] : i in [1..n div 2]]);
end function;
```

   Check independent points using the N-th Weil pairing `WeilPairing(P,Q, N);`

3. Modular polynomials are polynomials $\Phi_N(X, Y) \in \mathbb{Z}[X, Y]$ with the following property for E_1, E_2:

   - there exists a cyclic N-isogeny $E_1 \to E_2$ if and only if $\Phi_N(j(E_1), j(E_2)) = 0$ in $\mathbb{F}_q$;

   Note that this isogeny is not necessarily defined over $\mathbb{F}_q$. These polynomials are symmetric in X and Y; for $N = \ell$ prime, they have degree $\ell + 1$ in both X and Y. Moreover, they have giant coefficients, see Sutherland's database. Don't forget to reduce the modular polynomials to the finite field you are working in!

   Modular polynomials can be used to show there is an ℓ-isogeny between two elliptic curves, such as when you are defining the isogeny graphs. But they do not help you find the isogeny, compare this with Exercise (2).

   Self-loops in the supersingular isogeny graph correspond to self-isogenous elliptic curves, hence to roots of $\Phi_\ell(X, X)$. Note that this polynomial factors over $\mathbb{Z}$ already as a product of Hilbert class polynomials for imaginary quadratic fields that admit an element of norm ℓ. We will revisit this once we talk about endomorphisms of elliptic curves. For now, you can take as granted that

   - a self-ℓ-isogeny $\leftrightarrow$
   - endomorphism of degree ℓ $\leftrightarrow$
   - element in the endomorphism ring of norm ℓ $\leftrightarrow$
   - the endomorphism ring containing an order $\mathcal{O}$ in an imaginary quadratic field such that the order contains an element of norm ℓ, with p being inert in the imaginary quadratic field and $j(E)$ being the root mod $\mathbb{F}_{p^2}$ of the Hilbert class polynomial of $\mathcal{O}$.

   This is CM theory at its most beautiful! See more in Section 5.3.4 of Charles-Goren-Lauter .

   Magma commands: `ClassicalModularPolynomial` is the database of modular polynomials, be careful with the large coefficients.

```
R<X, Y>:=PolynomialRing(F, 2);   // F the finite field you need to define
Phi3 :=R!ClassicalModularPolynomial(3); Phi3;     // manageable coefficients
```

   You can evaluate polynomials using `Evaluate(poly, arg)` with arguments arg as a tuple. You can factor polynomials `Factorization`;

   You can find roots of polynomials using `Roots`; be careful that you need to do this for a polynomial in one variable.