Implementing cutting-edge isogeny-based cryptography for beginners

Lorenz Panny[1]   Krijn Reijnders[2]

[1] Technische Universität München

[2] Radboud University, Nijmegen

Summer School on Real-World Crypto and Privacy

Vodice, 7 June 2024

SQIsign: What?

https://sqisign.org

SQIsign: What?

https://sqisign.org

- A new and very hot post-quantum signature scheme.
- Based on an old and super cool part of mathematics. ⌣̈

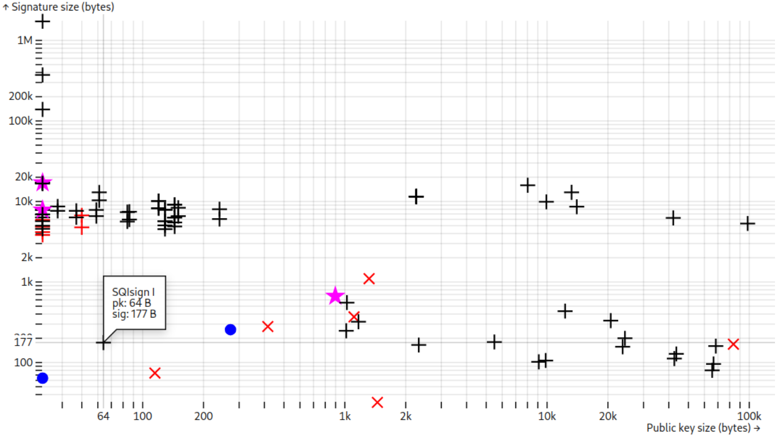
SQIsign: Why?

- \+ It's extremely small compared to the competition.
- – It's relatively slow compared to the competition.
- \+ ...but performance is getting better by the $\approx$ week!

SQIsign: Why?

+ It's extremely <u>small</u> compared to the competition.
– It's relatively <u>slow</u> compared to the competition.
+ ...but performance is getting better by the $\approx$ week!

SQIsign: Size comparisons



Source: `https://pqshield.github.io/nist-sigs-zoo`

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

$\rightsquigarrow$ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

$\rightsquigarrow$ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$E_0 \overset{\textit{secret}}{\dashrightarrow} E_A$$

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

⇝ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$\begin{tikzcd} E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_A \\ E_1 & \end{tikzcd}$$

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

⇝ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "secret"] \arrow[d, dashed, "commitment"'] & E_A \\
E_1 \arrow[r, "challenge"'] & E_2
\end{tikzcd}$$

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

⇝ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$\begin{tikzcd}
E_0 \arrow[rrr, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & & & E_A \arrow[d, thick, "\textit{response}"] \\
E_1 \arrow[rrr, "\textit{challenge}"'] & & & E_2
\end{tikzcd}$$

SQIsign: How?

- Paths in an "isogeny graph" seem hard to recover.

⇝ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$
\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & E_A \arrow[d, "\textit{response}"] \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}
$$

- Easy response: $E_A \to E_0 \to E_1 \to E_2$. *Obviously broken.*

SQIsign: How?

- Paths in an “isogeny graph” seem hard to recover.

⇝ Fiat–Shamir: signature scheme from identification scheme by replacing the challenger by a hash function.

$$\begin{tikzcd}
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & E_A \arrow[d, "\textit{response}"] \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}$$

- Easy response: $E_A \to E_0 \to E_1 \to E_2$. *Obviously broken.*
- **SQIsign**'s solution: Construct new path $E_A \to E_2$ (using *secret*).

The Deuring correspondence

...is a strong connection between two *a priori* very different worlds:

The Deuring correspondence

...is a strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.

The Deuring correspondence

...is a strong connection between two *a priori* very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

The Deuring correspondence

...is a strong connection between two $\overset{\textit{a priori}}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become "connecting ideals" in quaternion land.

The Deuring correspondence

...is a strong connection between two $\overset{a\ priori}{\curlyvee}$ very different worlds:

- Supersingular elliptic curves defined over $\mathbb{F}_{p^2}$.
- Quaternions: Maximal orders in a certain algebra $B_{p,\infty}$.

Isogenies become "connecting ideals" in quaternion land.

☺ One direction is easy, the other seems hard! $\rightsquigarrow$ *Cryptography!*

The Deuring correspondence (examples)

Let $p = 7799999$ and let $\mathbf{i}, \mathbf{j}$ satisfy $\mathbf{i}^2 = -1,\ \mathbf{j}^2 = -p,\ \mathbf{ji} = -\mathbf{ij}$.

The ring $\mathcal{O}_0 = \mathbb{Z} \oplus \mathbb{Z}\,\mathbf{i} \oplus \mathbb{Z}\,\frac{\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{1+\mathbf{ij}}{2}$
corresponds to the curve $E_0\colon\ y^2 = x^3 + x$.

The ring $\mathcal{O}_1 = \mathbb{Z} \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+32631010\mathbf{i}+\mathbf{ij}}{9894}$
corresponds to the curve $E_1\colon\ y^2 = x^3 + 1$.

The ideal $I = \mathbb{Z}\,4947 \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{598+4947\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947+598\mathbf{i}+\mathbf{ij}}{2}$
defines an isogeny $E_0 \to E_1$ of degree $4947 = 3 \cdot 17 \cdot 97$.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.
- Project the response (signature) down to the curve world.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.
- Project the response (signature) down to the curve world.
- The verifier can check on curves that everything is correct.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.
- Project the response (signature) down to the curve world.
- The verifier can check on curves that everything is correct.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$ and find elements in $\mathrm{Hom}(E, E')$ that are easy to compute.

SQIsign

<u>Main idea:</u>

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.
- Project the response (signature) down to the curve world.
- The verifier can check on curves that everything is correct.

<u>Main technical tool:</u> The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$ and find elements in $\mathrm{Hom}(E, E')$ that are easy to compute.

$\rightsquigarrow$ SQIsign takes the “broken” signature $E_A \to E_0 \to E_1 \to E_2$ and rewrites it into a random path $E_A \to E_2$.

SQIsign

Main idea:

- Translate $E_A \to E_0 \to E_1 \to E_2$ to quaternion land.
- Compute a random path $E_A \to E_2$ using quaternion magic.
- Project the response (signature) down to the curve world.
- The verifier can check on curves that everything is correct.

Main technical tool: The KLPT algorithm.

- From $\mathrm{End}(E), \mathrm{End}(E')$, can randomize within $\mathrm{Hom}(E, E')$ and find elements in $\mathrm{Hom}(E, E')$ that are easy to compute.

$\rightsquigarrow$ SQIsign takes the “broken” signature $E_A \to E_0 \to E_1 \to E_2$ and rewrites it into a random path $E_A \to E_2$.

> *“If you have KLPT implemented very nicely as a black box, then anyone can implement SQIsign.”*
>
> — Yan Bo Ti

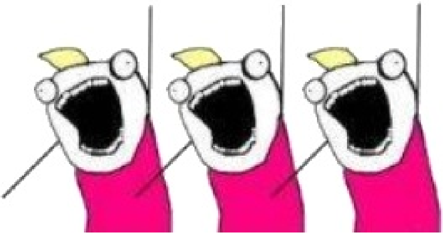
SQIsign: Where?

SQIsign: Where?

TODO: insert picture of beach in Croatia or whatever, idk

SQIsign: When?

SQIsign: When?



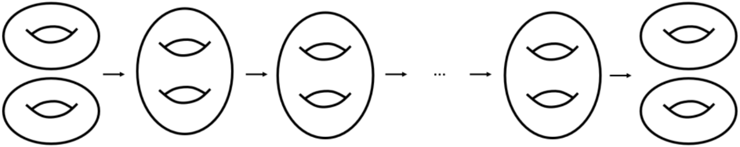
The HD situation

The HD situation

- The “SIKE attacks” of Summer 2022 have sparked a revolution in isogeny-based cryptography.

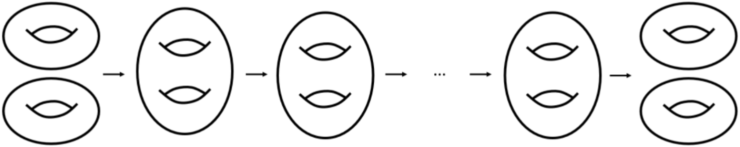
The HD situation

- The "SIKE attacks" of Summer 2022 have sparked a revolution in isogeny-based cryptography.
- Computing isogenies of higher-dimensional abelian varieties is a very powerful tool: Also for SQIsign!



The HD situation

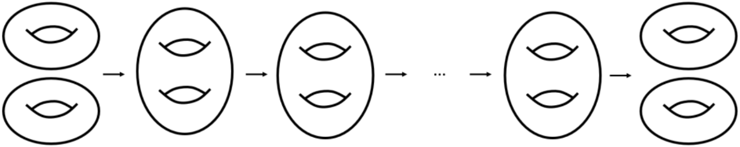
- The "SIKE attacks" of Summer 2022 have sparked a revolution in isogeny-based cryptography.
- Computing isogenies of higher-dimensional abelian varieties is a very powerful tool: Also for SQIsign!



- Recent preprints: Faster using 2-dimensional isogenies.

The HD situation

- The “SIKE attacks” of Summer 2022 have sparked a revolution in isogeny-based cryptography.
- Computing isogenies of higher-dimensional abelian varieties is a very powerful tool: Also for SQIsign!



- Recent preprints: Faster using 2-dimensional isogenies.
- Here, we only talk about the 1-dimensional approach: Easier for beginners, and still at the heart of everything.

The plan

- You'll get some prepared Python scripts.
- Some parts are missing. Fill them in.
- For the rest of this talk we explain the underlying math.

The plan

- You'll get some prepared Python scripts.
- Some parts are missing. Fill them in.
- For the rest of this talk we explain the underlying math.

⚠ We're doing everything in Python for the moment,
so secure implementations are out of scope.

The plan

- You'll get some prepared Python scripts.
- Some parts are missing. Fill them in.
- For the rest of this talk we explain the underlying math.

⚠️ We're doing everything in Python for the moment,
so secure implementations are out of scope.

- Optional **challenge**: Minimize the "cost" for your implementation: <u>Estimated</u> equivalent # multiplications.

The plan

- You'll get some prepared Python scripts.
- Some parts are missing. Fill them in.
- For the rest of this talk we explain the underlying math.

⚠️ We're doing everything in Python for the moment,
so secure implementations are out of scope.

- Optional **challenge**: Minimize the "cost" for your implementation: Estimated equivalent # multiplications.
- There will be ***prizes!!*** 🎉

Stand back!

We're going to do math.

Some of the coming bits are on the math-heavy side, but don't worry: Things will be very concrete at the end.

Elliptic curves (special case)

An elliptic curve for our purposes over a field F is an equation

$$E\colon\ y^2 = x^3 + Ax^2 + x$$

with $A \in F \setminus \{\pm 2\}$.

Elliptic curves (special case)

An elliptic curve for our purposes over a field F is an equation

$$E\colon\; y^2 = x^3 + Ax^2 + x$$

with $A \in F \setminus \{\pm 2\}$.

A point on E is a solution (x, y), <u>or</u> the "fake" point ∞.

Elliptic curves (special case)

An elliptic curve for our purposes over a field F is an equation

$$E\colon\ y^2 = x^3 + Ax^2 + x$$

with $A \in F \setminus \{\pm 2\}$.

A point on E is a solution (x, y), or the "fake" point ∞.

E is an abelian group: we can "add" points.

Elliptic curves (special case)

An elliptic curve for our purposes over a field F is an equation

$$E\colon\ y^2 = x^3 + Ax^2 + x$$

with $A \in F \setminus \{\pm 2\}$.

A point on E is a solution (x, y), <u>or</u> the "fake" point ∞.

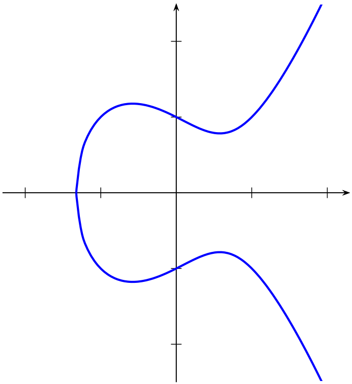
E is an abelian group: we can "add" points.

- The neutral element is ∞.
- The inverse of (x, y) is $(x, -y)$.
- The sum of (x_1, y_1) and (x_2, y_2) is

$$\left(\lambda^2 - A - x_1 - x_2,\ \lambda(2x_1 + x_2 + A - \lambda^2) - y_1\right)$$

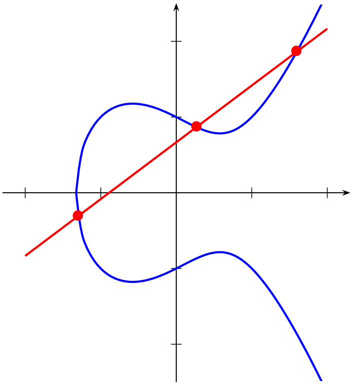
where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $x_1 \neq x_2$ and $\lambda = \frac{3x_1^2 + 2Ax + 1}{2y_1}$ otherwise.

Elliptic curves (picture over $\mathbb{R}$)



An elliptic curve $\mathbb{R}$.

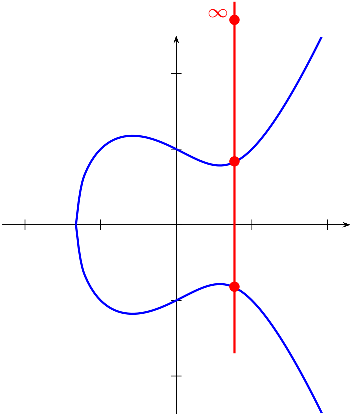
Elliptic curves (picture over $\mathbb{R}$)



<u>Addition law</u>:

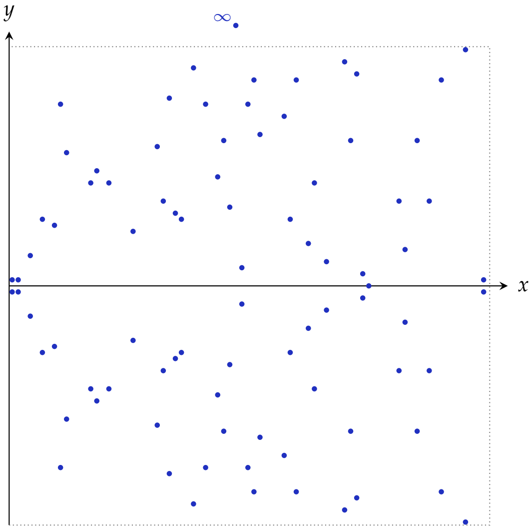
$P + Q + R = \infty \iff \{P, Q, R\}$ on a straight line.

Elliptic curves (picture over $\mathbb{R}$)



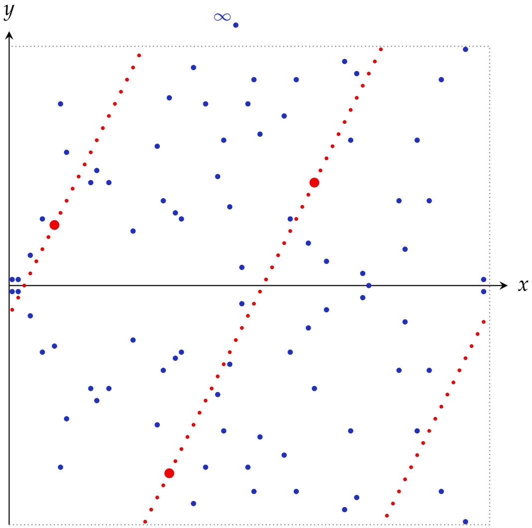
The *point at infinity* ∞ lies on every vertical line.

Elliptic curves (picture over $\mathbb{F}_p$)



The same curve as before over the finite field $\mathbb{F}_{79}$.

Elliptic curves (picture over $\mathbb{F}_p$)



The addition law over the finite field $\mathbb{F}_{79}$.

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

Define scalar multiplication $[n]P := \underbrace{P + \cdots + P}_{n \text{ times}}$. (Use $O(\log n)$ algorithm!)

ECDH (not post-quantum)

Public parameters:
an elliptic curve E and a point $P \in E$ of large prime order ℓ.

Define scalar multiplication $[n]P := \underbrace{P + \cdots + P}_{n \text{ times}}$. (Use $O(\log n)$ algorithm!)

Alice	public	Bob
$a \xleftarrow{\text{random}} \{0 \ldots \ell-1\}$		$b \xleftarrow{\text{random}} \{0 \ldots \ell-1\}$
$[a]P$	$\longrightarrow$ $[a]P$ to Bob; $[b]P$ to Alice $\longleftarrow$	$[b]P$
$s := [a]([b]P)$	$\longleftrightarrow$ equal!	$s := [b]([a]P)$

Projective coordinates

Computationally, it usually makes sense to trade inversions for (a few) multiplications and additions.

Projective coordinates

Computationally, it usually makes sense to trade inversions for (a few) multiplications and additions.

Hence, instead of storing $u, v \in F$, store $u_1, u_2, v_1, v_2 \in F$ where $u = u_1/u_2$ and $v = v_1/v_2$.

Projective coordinates

Computationally, it usually makes sense to trade inversions for (a few) multiplications and additions.

Hence, instead of storing $u, v \in F$, store $u_1, u_2, v_1, v_2 \in F$ where $u = u_1/u_2$ and $v = v_1/v_2$.

- Inverting u_1/u_2 just means swapping u_1 and u_2.
- $u_1/u_2 \cdot v_1/v_2$ means computing $u_1 v_1$ and $u_2 v_2$.
- $u_1/u_2 + v_1/v_2$ means computing $u_1 v_2 + u_2 v_1$ and $u_2 v_2$.

Projective coordinates

Computationally, it usually makes sense to trade inversions for (a few) multiplications and additions.

Hence, instead of storing $u, v \in F$, store $u_1, u_2, v_1, v_2 \in F$ where $u = u_1/u_2$ and $v = v_1/v_2$.

- Inverting u_1/u_2 just means swapping u_1 and u_2.
- $u_1/u_2 \cdot v_1/v_2$ means computing $u_1 v_1$ and $u_2 v_2$.
- $u_1/u_2 + v_1/v_2$ means computing $u_1 v_2 + u_2 v_1$ and $u_2 v_2$.

For elliptic-curve points, we commonly work with $(X : Y : Z)$ representing $(x, y) = (X/Z, Y/Z)$. Then actually $\infty = (0 : 1 : 0)$!

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\Longrightarrow$ We get an induced map $\mathtt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \mathtt{xMUL}_n(x(P)) = x([n]P)\,.$$

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\implies$ We get an induced map $\mathtt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \mathtt{xMUL}_n(x(P)) = x([n]P)\,.$$

Here again, we often use projective coordinates (X, Z) representing the x-coordinate X/Z.

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\Longrightarrow$ We get an induced map $\mathtt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \mathtt{xMUL}_n(x(P)) = x([n]P)\,.$$

Here again, we often use projective coordinates (X, Z) representing the x-coordinate X/Z.

"Correct" technical meaning: The x-coordinate projection is a map to the Kummer line $\mathcal{K}$, which is basically just $\mathbb{P}^1$.

Special case: $\infty \mapsto (1:0) \in \mathcal{K}$.

x-only arithmetic

Observation:

- For $n \in \mathbb{Z}$, we have $[n](-P) = -[n]P$. (This holds in any group.)
- Recall that $P = (x, y)$ has inverse $-P = (x, -y)$.

$\Longrightarrow$ We get an induced map $\mathtt{xMUL}_n$ on x-coordinates such that

$$\forall P \in E. \quad \mathtt{xMUL}_n(x(P)) = x([n]P)\,.$$

Here again, we often use projective coordinates (X, Z) representing the x-coordinate X/Z.

"Correct" technical meaning: The x-coordinate projection is a map to the Kummer line $\mathcal{K}$, which is basically just $\mathbb{P}^1$.

Special case: $\infty \mapsto (1:0) \in \mathcal{K}$.

For many *fun* facts about Kummer varieties, ask Krijn.

x-only arithmetic: Building blocks

There are algebraic formulas for the following operations:

- `xDBL`: maps $x(P)$ to $x([2]P)$.
- `xADD`: maps $x(P), x(Q), x(P-Q)$ to $x(P+Q)$.

x-only arithmetic: Building blocks

There are algebraic formulas for the following operations:

- `xDBL`: maps $x(P)$ to $x([2]P)$.
- `xADD`: maps $x(P), x(Q), x(P-Q)$ to $x(P+Q)$.

Fast formulas may be found here:

`https://hyperelliptic.org/EFD/g1p/auto-montgom-xz.html`

x-only arithmetic: Building blocks

There are algebraic formulas for the following operations:

- `xDBL`: maps $x(P)$ to $x([2]P)$.
- `xADD`: maps $x(P), x(Q), x(P-Q)$ to $x(P+Q)$.

Fast formulas may be found here:

`https://hyperelliptic.org/EFD/g1p/auto-montgom-xz.html`

⚠ In these formulas, the projectivized x-coordinates of the points P, Q, $P-Q$ are `(X2,Z2)`, `(X3,Z3)`, `(X1,Z1)`!

x-only arithmetic: The Montgomery ladder

- Recall: Points on $\mathcal{K}$ are pairs $(X : Z)$ of finite-field elements representing the x-coordinate X/Z, or ∞ if $Z = 0$.

x-only arithmetic: The Montgomery ladder

- Recall: Points on $\mathcal{K}$ are pairs $(X : Z)$ of finite-field elements representing the x-coordinate X/Z, or ∞ if $Z = 0$.

<u>Input:</u> Integer $n \in \mathbb{Z}$, Kummer line point $\pm P \in \mathcal{K}$.
<u>Output:</u> The Kummer line point $\mathtt{xMUL}_n(\pm P) = \pm[n]P$.

1. If $n < 0$, set $n := -n$.
2. Binary expansion: $n = \sum_{i=0}^{\ell-1} b_i 2^i$ with each $b_i \in \{0, 1\}$.
3. Initialize $\pm R_0 := (1 : 0) \in \mathcal{K}$ and $\pm R_1 := \pm P \in \mathcal{K}$.
4. For k ranging from $\ell - 1$ down to 0:
    - Set $\pm R_{1-b_k} := \mathtt{xADD}(\pm R_0, \pm R_1, \pm P)$.
    - Set $\pm R_{b_k} := \mathtt{xDBL}(\pm R_{b_k})$.
5. Return $\pm R_0$.

x-only arithmetic: The Montgomery ladder

- Recall: Points on $\mathcal{K}$ are pairs $(X : Z)$ of finite-field elements representing the x-coordinate X/Z, or ∞ if $Z = 0$.

<u>Input:</u> Integer $n \in \mathbb{Z}$, Kummer line point $\pm P \in \mathcal{K}$.
<u>Output:</u> The Kummer line point $\mathtt{xMUL}_n(\pm P) = \pm[n]P$.

1. If $n < 0$, set $n := -n$.
2. Binary expansion: $n = \sum_{i=0}^{\ell-1} b_i 2^i$ with each $b_i \in \{0, 1\}$.
3. Initialize $\pm R_0 := (1 : 0) \in \mathcal{K}$ and $\pm R_1 := \pm P \in \mathcal{K}$.
4. For k ranging from $\ell - 1$ down to 0:
   - Set $\pm R_{1-b_k} := \mathtt{xADD}(\pm R_0, \pm R_1, \pm P)$.
   - Set $\pm R_{b_k} := \mathtt{xDBL}(\pm R_{b_k})$.
5. Return $\pm R_0$.

- Loop invariant: $\pm(R_1 - R_0) = \pm P$, and $\pm R_0 = \pm[n \gg k]P$.

Isogenies

Isogenies

...are just fancily-named

nice maps

between elliptic curves.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Reminder:

A rational function is $f(x, y)/g(x, y)$ where f, g are polynomials.

A group homomorphism φ satisfies $\varphi(P + Q) = \varphi(P) + \varphi(Q)$.

Isogenies

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Reminder:

A rational function is $f(x,y)/g(x,y)$ where f, g are polynomials.

A group homomorphism φ satisfies $\varphi(P+Q) = \varphi(P) + \varphi(Q)$.

The kernel of an isogeny $\varphi\colon E \to E'$ is $\{P \in E : \varphi(P) = \infty\}$.

The degree of a separable* isogeny is the size of its kernel.

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #1: $(x, y) \mapsto \left(\frac{x^3-4x^2+30x-12}{(x-2)^2}, \frac{x^3-6x^2-14x+35}{(x-2)^3} \cdot y \right)$
defines a degree-3 isogeny of the elliptic curves

$$\{y^2 = x^3 + x\} \longrightarrow \{y^2 = x^3 - 3x + 3\}$$

over $\mathbb{F}_{71}$. Its kernel is $\{(2, 9), (2, -9), \infty\}$.

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #2: For any a and b, the map $\iota\colon (x,y) \mapsto (-x, \sqrt{-1}\cdot y)$ defines a degree-1 isogeny of the elliptic curves

$$\{y^2 = x^3 + ax + b\} \longrightarrow \{y^2 = x^3 + ax - b\}\,.$$

It is an *isomorphism*; its kernel is $\{\infty\}$.

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #3: For each $m \neq 0$, the multiplication-by-m map

$$[m] \colon E \to E$$

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #3: For each $m \neq 0$, the multiplication-by-m map

$$[m] \colon E \to E$$

is a degree-m^2 isogeny. If $m \neq 0$ in the base field, its kernel is

$$E[m] \;\cong\; \mathbb{Z}/m \times \mathbb{Z}/m.$$

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #4: For $E/\mathbb{F}_q$, the map

$$\pi\colon (x, y) \mapsto (x^q, y^q)$$

is a degree-q isogeny, the *Frobenius endomorphism*.

Isogenies (examples)

An isogeny of elliptic curves is a non-zero map $E \to E'$ that is:

- given by rational functions.
- a group homomorphism.

Example #4: For $E/\mathbb{F}_q$, the map

$$\pi : (x, y) \mapsto (x^q, y^q)$$

is a degree-q isogeny, the *Frobenius endomorphism*.

The kernel of $\pi - 1$ is precisely the set of rational points $E(\mathbb{F}_q)$.

The isogeny relation

Isogenies between distinct curves are "rare".

We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

The isogeny relation

Isogenies between distinct curves are “rare”.
We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi\colon E \to E'$ has a unique dual isogeny $\widehat{\varphi}\colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

The isogeny relation

Isogenies between distinct curves are "rare".
We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi\colon E \to E'$ has a unique dual isogeny $\widehat{\varphi}\colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

Tate's theorem:
$E, E'/\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

The isogeny relation

Isogenies between distinct curves are "rare".
We say E and E' are *isogenous* if there exists an isogeny $E \to E'$.

Each isogeny $\varphi\colon E \to E'$ has a unique dual isogeny $\widehat{\varphi}\colon E' \to E$ characterized by $\widehat{\varphi} \circ \varphi = [\deg \varphi]$ and $\varphi \circ \widehat{\varphi} = [\deg \varphi]$.

Tate's theorem:
$E, E'/\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.

$\Longrightarrow$ Bottom line: Being isogenous is an equivalence relation.

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

[1] (up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G \colon E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
  (...but they are only efficient for "small" degrees!)

[1](up to isomorphism of E')

Isogenies and kernels

For any finite subgroup G of E, there exists a unique[1] separable* isogeny $\varphi_G : E \to E'$ with kernel G.

The curve E' is denoted by E/G. (cf. quotient groups)

If G is defined over k, then φ_G and E/G are also defined over k.

$\rightsquigarrow$ To choose an isogeny, simply choose a finite subgroup.

- We have formulas to compute and evaluate isogenies.
(...but they are only efficient for "small" degrees!)

$\rightsquigarrow$ Decompose large-degree isogenies into prime steps.
That is: Walk in an isogeny graph.

[1](up to isomorphism of E')

Outgoing isogenies: How many and which ones?

- By the isogeny–subgroup correspondence, every step in the ℓ-isogeny graph comes from a subgroup of size ℓ.

Outgoing isogenies: How many and which ones?

- By the isogeny–subgroup correspondence, every step in the ℓ-isogeny graph comes from a subgroup of size ℓ.

$\rightsquigarrow$ For $\ell = 2$, these are exactly points of order 2.

Outgoing isogenies: How many and which ones?

- By the isogeny–subgroup correspondence, every step in the ℓ-isogeny graph comes from a subgroup of size ℓ.

$\rightsquigarrow$ For $\ell = 2$, these are exactly points of order 2.

- Recall $\ker[m] \cong \mathbb{Z}/m \times \mathbb{Z}/m$ for if $p \nmid m$.
  In other words: $\exists P, Q \in E[m]$ such that

$$\{0, \dots, m-1\}^2 \to E,\ (i, j) \mapsto [a]P + [b]Q$$

  is a bijection. We call (P, Q) an m-torsion basis.

Outgoing isogenies: How many and which ones?

- By the isogeny–subgroup correspondence, every step in the ℓ-isogeny graph comes from a subgroup of size ℓ.

$\rightsquigarrow$ For $\ell = 2$, these are exactly points of order 2.

- Recall $\ker[m] \cong \mathbb{Z}/m \times \mathbb{Z}/m$ for if $p \nmid m$.
  In other words: $\exists P, Q \in E[m]$ such that

$$\{0, \ldots, m-1\}^2 \to E,\ (i,j) \mapsto [a]P + [b]Q$$

  is a bijection. We call (P, Q) an *m*-torsion basis.

$\rightsquigarrow$ For $m = 2$ there are three outgoing 2-isogenies:
They have kernels $(\alpha, 0)$ where α is a root of $x^3 + Ax^2 + x$.

Outgoing isogenies: How many and which ones?

- By the isogeny–subgroup correspondence, every step in the ℓ-isogeny graph comes from a subgroup of size ℓ.

$\rightsquigarrow$ For $\ell = 2$, these are exactly points of order 2.

- Recall $\ker[m] \cong \mathbb{Z}/m \times \mathbb{Z}/m$ for if $p \nmid m$.
  In other words: $\exists P, Q \in E[m]$ such that

$$\{0, ..., m-1\}^2 \to E,\ (i, j) \mapsto [a]P + [b]Q$$

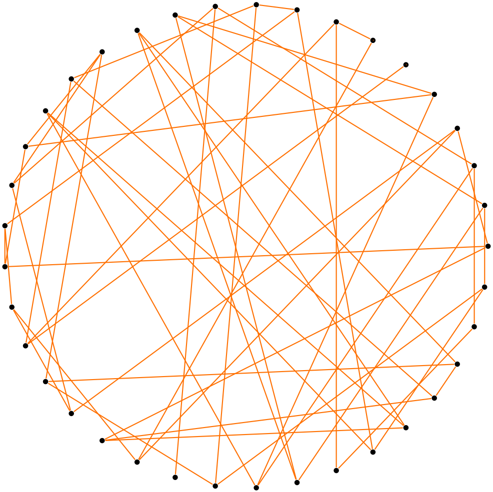
is a bijection. We call (P, Q) an m-torsion basis.

$\rightsquigarrow$ For $m = 2$ there are three outgoing 2-isogenies:
They have kernels $(\alpha, 0)$ where α is a root of $x^3 + Ax^2 + x$.

$\rightsquigarrow$ For $m = 2^n$ there are $3 \cdot 2^{n-1}$ outgoing 2^n-isogenies:
They are given by $\langle P + [s]Q \rangle$ and $\langle [r]P + Q \rangle$ with r even.

The (supersingular) 2-isogeny graph

Over $\mathbb{F}_{p^2}$:



Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \Bigg(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \\ y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \Bigg)$$

defines an isogeny of elliptic curves with kernel G.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \Bigg(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \\ y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \Bigg)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

Computing isogenies: Vélu's formulas (1971)

Let G be a finite subgroup of an elliptic curve E. Then

$$P \mapsto \left(x(P) + \sum_{Q \in G \setminus \{\infty\}} \big(x(P+Q) - x(Q) \big), \right.$$
$$\left. y(P) + \sum_{Q \in G \setminus \{\infty\}} \big(y(P+Q) - y(Q) \big) \right)$$

defines an isogeny of elliptic curves with kernel G.

This leads to formulas for

- computing the defining equation of E/G;
- evaluating the isogeny $E \to E/G$ at a point.

<u>Complexity</u>: $\Theta(\#G)$ $\rightsquigarrow$ only suitable for small degrees.

The $\sqrt{\text{é}}$lu algorithm reduces the cost to $\widetilde{\mathcal{O}}(\sqrt{\#G})$.

Some 2-isogeny formulas

Proposition 1. *Let K be a field with $\mathrm{char}(K) \neq 2$. Let $a \in K$ such that $a^2 \neq 4$ and $E/K : y^2 = x^3 + ax^2 + x$ is a Montgomery curve. Let $G \subset E(\bar{K})$ be a finite subgroup such that $(0,0) \notin G$ and let ϕ be a separable isogeny such that $\ker(\phi) = G$. Then there exists a curve $\widetilde{E}/K : y^2 = x^3 + Ax^2 + x$ such that, up to post-composition by an isomorphism,*

$$\begin{aligned} \phi : E &\to \widetilde{E} \\ (x, y) &\mapsto (f(x), c_0 y f'(x)) \end{aligned}$$

where

$$f(x) = x \prod_{T \in G \setminus \{\mathcal{O}_E\}} \frac{x x_T - 1}{x - x_T} .$$

Moreover, writing

$$\pi = \prod_{T \in G \setminus \{\mathcal{O}_E\}} x_T , \qquad \sigma = \sum_{T \in G \setminus \{\mathcal{O}_E\}} \left(x_T - \frac{1}{x_T} \right) ,$$

we have that $A = \pi(a - 3\sigma)$ and $c_0^2 = \pi$.

Source: Joost Renes: “Computing isogenies between Montgomery curves using the action of $(0,0)$”, 2017

Some 2-isogeny formulas

Simplified for your convenience:

$$\varphi_K\colon\ \left\{y^2 = x^3 + Ax^2 + x\right\} \longrightarrow \left\{y^2 = x^3 + \big(Ax_K - 3(x_K^2 - 1)\big)x^2 + x\right\}$$
$$(x,\ \ldots) \longmapsto \left(\tfrac{x(xx_K-1)}{(x-x_K)},\ \ldots\right)$$

A specific 2-isogeny formula

Remark 6. In [FJP14, §4.3.2] the authors describe a 2-isogeny with kernel (0,0) as

$$\varphi : E \to F : by^2 = x^3 + (a+6)x^2 + 4(2+a)x$$
$$(x, y) \mapsto \left(\frac{(x-1)^2}{x}, y\left(1 - \frac{1}{x^2}\right) \right) .$$

The coefficient of x can be removed by computing $2\sqrt{a+2}$ and composing with the isomorphism

$$(x, y) \mapsto \left(\frac{x}{2\sqrt{a+2}}, \frac{y}{2\sqrt{a+2}} \right) ,$$

putting F in the desired form. This requires computing a square root, which could be avoided by having knowledge of a point $P_8 = \left(2\sqrt{a+2}, —\right)$ of order 8 above $(0,0)$.

Source: Joost Renes: “Computing isogenies between Montgomery curves using the action of $(0, 0)$”, 2017

A specific 2-isogeny formula

Simplified for your convenience:

$$\begin{aligned}\varphi_{(0,0)}\colon\ \{y^2 = x^3 + Ax^2 + x\} &\longrightarrow \{y^2 = x^3 + \tfrac{A+6}{2\sqrt{A+2}}x^2 + x\}\\ (x,\ \ldots) &\longmapsto \left(\tfrac{(x-1)^2}{2\sqrt{A+2}\cdot x},\ \ldots\right)\end{aligned}$$

SQIsign verification

Coming back to our diagram from earlier:

$$\begin{tikzcd}[column sep=huge, row sep=huge]
E_0 \arrow[r, dashed, "\textit{secret}"] \arrow[d, dashed, "\textit{commitment}"'] & E_A \arrow[d, thick, "\textit{response}"] \\
E_1 \arrow[r, "\textit{challenge}"'] & E_2
\end{tikzcd}$$

After Fiat–Shamir, the verification procedure is:

1. <u>Input</u>: Public key E_A, commitment curve E_1, description of the response isogeny (later).
2. Compute the response isogeny; call the result $E_2^{(resp)}$.
3. Recompute the challenge isogeny from E_1 and the message using a hash function; call the result $E_2^{(chall)}$.
4. Check that $E_2^{(resp)} = E_2^{(chall)}$ and that $2 \nmid \widehat{challenge} \circ response$.

SQIsign verification (isogeny chains)

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi \colon E \to E/\langle K \rangle$.

SQIsign verification (isogeny chains)

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K \rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

SQIsign verification (isogeny chains)

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi \colon E \to E/\langle K \rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

SQIsign verification (isogeny chains)

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K\rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(n^2 \cdot \ell)$.
Exponentially smaller than a ℓ^n-isogeny!

SQIsign verification (isogeny chains)

Main task in **SQIsign verification**:

Given E and $K \in E$ of order ℓ^n, compute $\psi\colon E \to E/\langle K\rangle$.

- Vélu's formulas take $\Theta(\ell^n)$ to compute ψ.

!! Evaluate ψ as a chain of small-degree isogenies:

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\psi"'] & E_1 \arrow[r, "\psi_2"] & \dots \arrow[r, "\psi_{n-1}"] & E_{n-1} \arrow[r, "\psi_n"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(n^2 \cdot \ell)$.
Exponentially smaller than a ℓ^n-isogeny!

- Graph view: Each ψ_i is a step in the ℓ-isogeny graph.

Predictable groups

Elliptic curves in general can be very annoying

Predictable groups

Elliptic curves in general can be very annoying *computationally*: Points in $E[\ell]$ have a tendency to live in large extension fields.

Predictable groups

Elliptic curves in general can be very annoying *computationally*:
Points in $E[\ell]$ have a tendency to live in large extension fields.

Solution:

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *supersingular* if and only if $\#E(\mathbb{F}_p) = p+1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ and
  $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p+1) \times \mathbb{Z}/(p+1)$.

Predictable groups

Elliptic curves in general can be very annoying *computationally*:
Points in $E[\ell]$ have a tendency to live in large extension fields.

Solution:

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *supersingular* if and only if $\#E(\mathbb{F}_p) = p+1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p+1)$ and
  $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p+1) \times \mathbb{Z}/(p+1)$.

$\rightsquigarrow$ Easy method to control the group structure by choosing p!

$\rightsquigarrow$ Cryptography works well using supersingular curves.

Predictable groups

Elliptic curves in general can be very annoying *computationally*: Points in $E[\ell]$ have a tendency to live in large extension fields.

<u>Solution:</u>

Let $p \geq 5$ be prime.

- $E/\mathbb{F}_p$ is *<u>supersingular</u>* if and only if $\#E(\mathbb{F}_p) = p{+}1$.
- In that case, $E(\mathbb{F}_p) \cong \mathbb{Z}/(p{+}1)$ and
  $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p{+}1) \times \mathbb{Z}/(p{+}1)$.

$\rightsquigarrow$ Easy method to control the group structure by choosing p!

$\rightsquigarrow$ Cryptography works well using supersingular curves.

(All "curves" are elliptic and have $E(\mathbb{F}_{p^2}) = E[p+1]$ for the rest of the day.)

Now: Your turn!

- First: Overview of the code structure.