p-adic attacks on elliptic curves

Overdrive Hacking Conference 2019

Enric Florit

Who am I?

Enric Florit

- Maths + CS student at University of Barcelona
- I do number theory and cryptography
- Member of Hacking Lliure

HACKING
LLIURE

Introduction

What are Elliptic Curves?

Elliptic Curve Cryptography

Attacking ECC

The p-adic attack

Public-key cryptography

We want to

- Exchange keys (for symmetric crypto)
- Sign data
- Encrypt things?

RSA

RSA is easier to understand, it's basic modular arithmetic:

- Take two **large** primes p and q, set $N = p \times q$ and $\varphi(N) = (p-1)(q-1)$.
- Choose e between 1 and $\varphi(pq)$ such that $\gcd(e, \varphi(pq)) = 1$.
- Compute $d \equiv e^{-1} \mod \varphi(pq)$.
- $pub = (N, e)$, $priv = (p, q, d)$.

The equation of an elliptic curve

An elliptic curve is the set of points (x, y) in the plane that satisfy the equation

$$y^2 = x^3 + Ax + B,$$

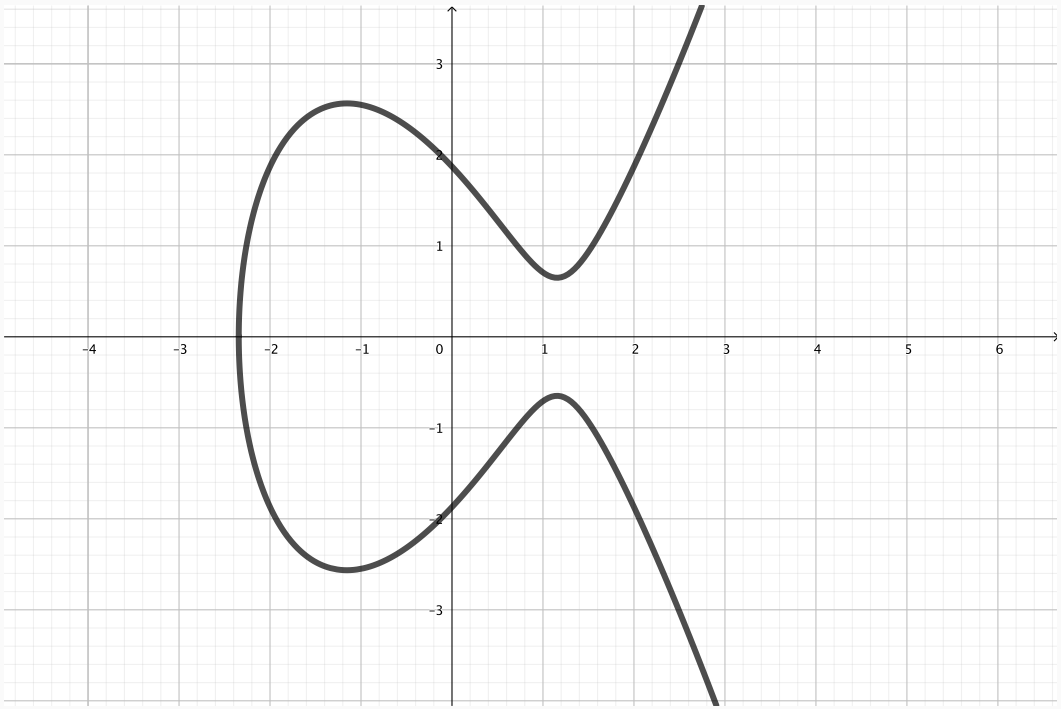
where A, B are constants[1] (integer, rational, real, complex...).

[1] With $4A^3 + 27B^2 \neq 0$

The graph of an elliptic curve

$$y^2 = x^3 - 4x + \frac{7}{2}$$

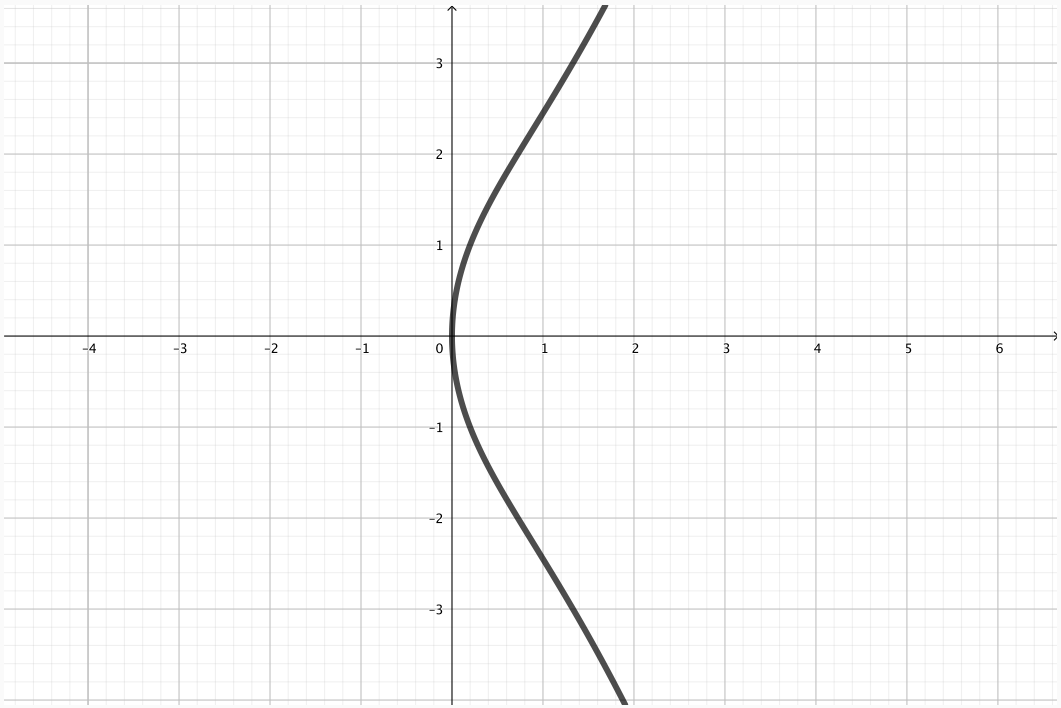
If we plot this curve, we have a picture like this:



The graph of an elliptic curve

$$y^2 = x^3 + 5x$$

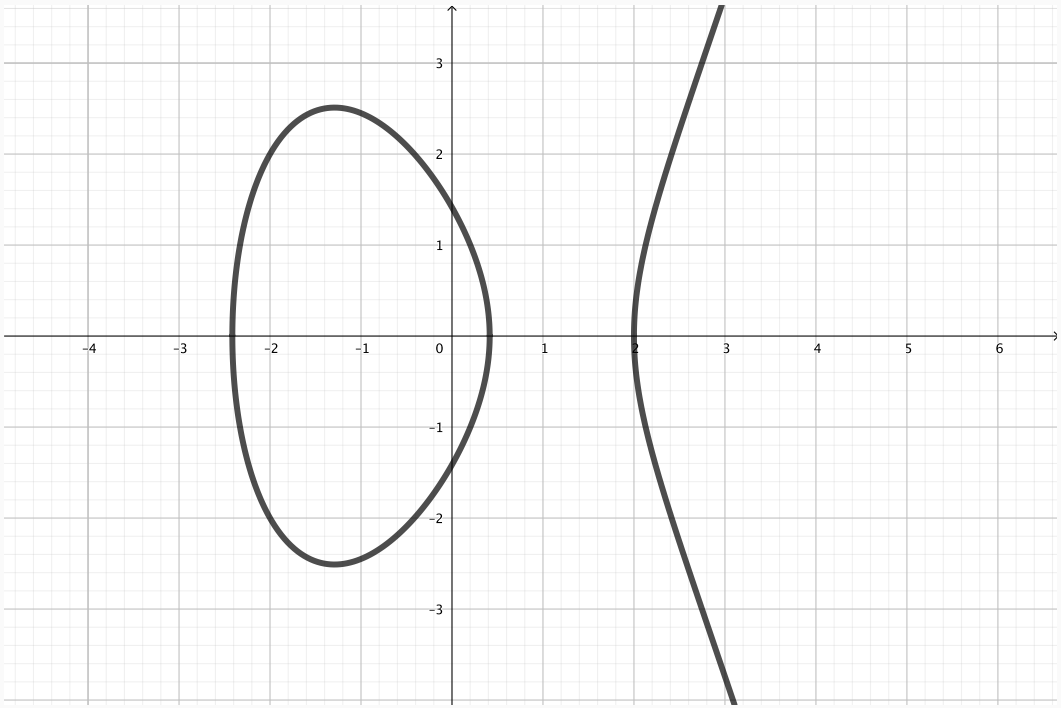
If we plot this curve, we have a picture like this:



The graph of an elliptic curve

$$y^2 = x^3 - 5x + 2$$

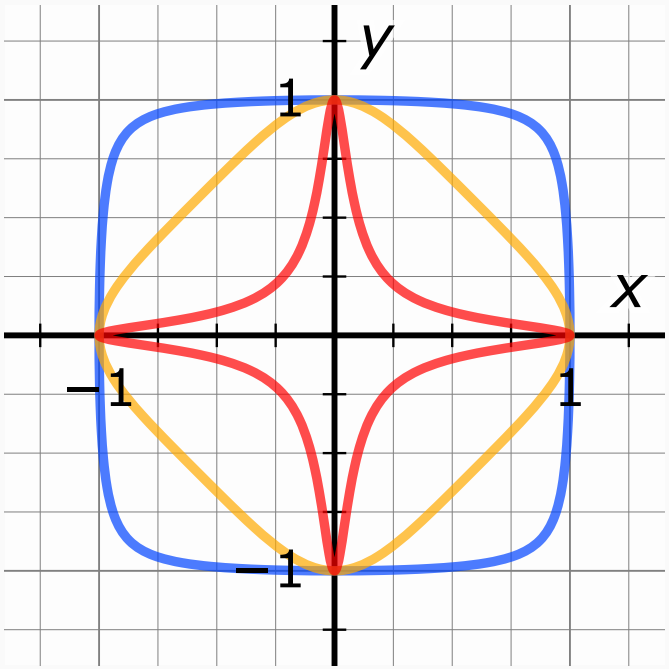
If we plot this curve, we have a picture like this:



There exist other equations

We could also work with other **equivalent** curves

$$x^2 + y^2 = 1 - dx^2y^2$$

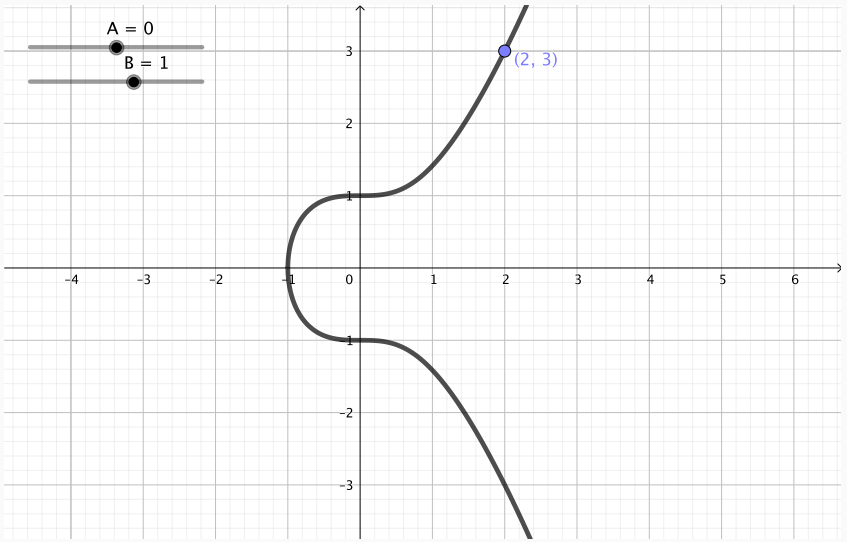


Edwards curves with $d = 300$ (red), $d = \sqrt{8}$ (yellow) and $d = -0.9$ (blue)[2]

[2] https://commons.wikimedia.org/wiki/File:Edward-curves.svg

The graph of an elliptic curve

We say a point (x, y) is on the curve if it satisfies the equation $y^2 = x^3 + Ax + B$.



For example $(2, 3)$ satisfies

$$\begin{cases} x^3 + 1 = 2^3 + 1 & = 9 \\ y^2 = 3^2 & = 9 \end{cases} \implies \mathbf{y^2 = x^3 + 1.}$$

Sum of points

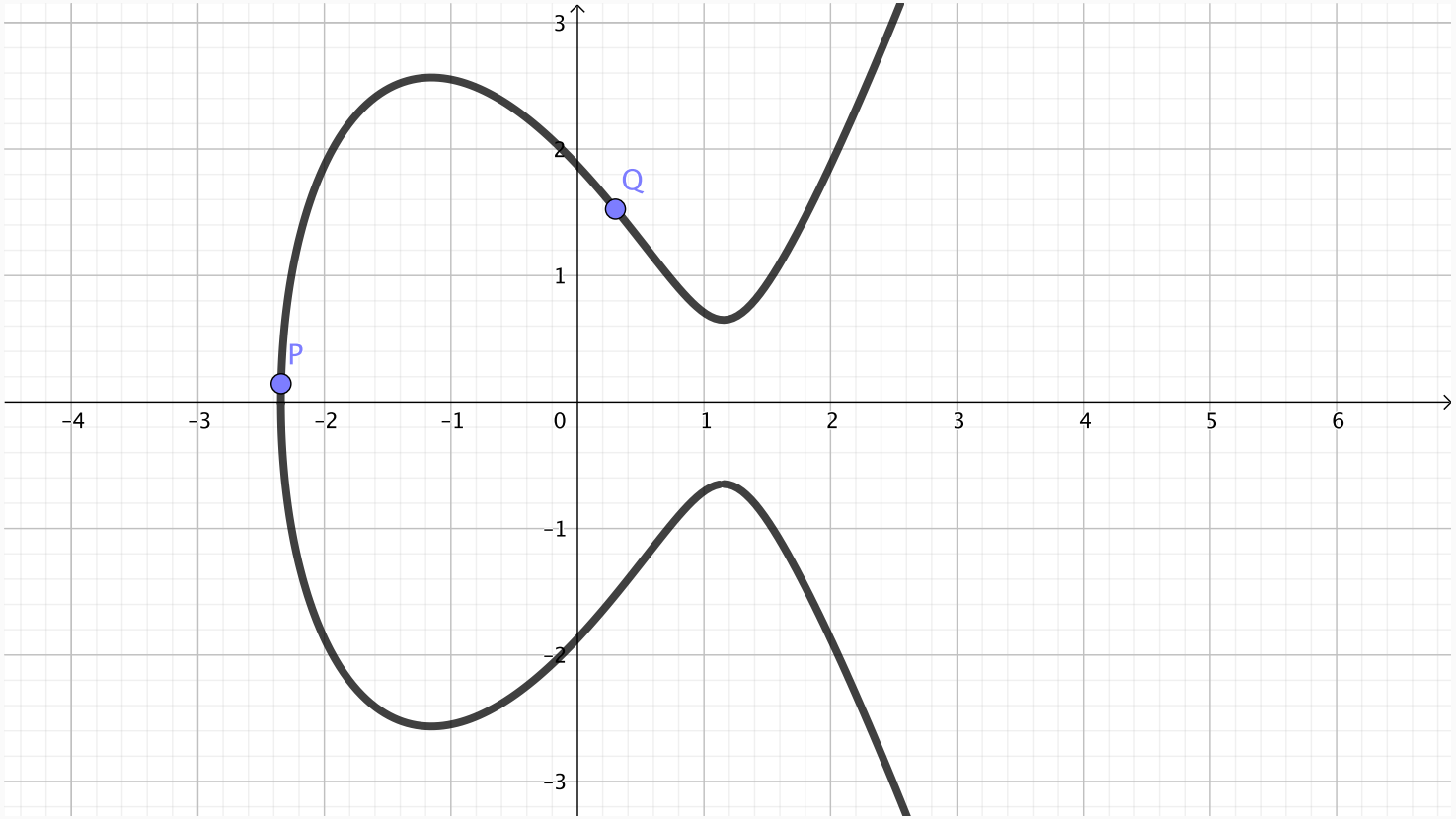
We care about elliptic curves because we can **add points**.

Huh? Coordinate-wise sum? No!

We are going to add pairs of points **geometrically**.

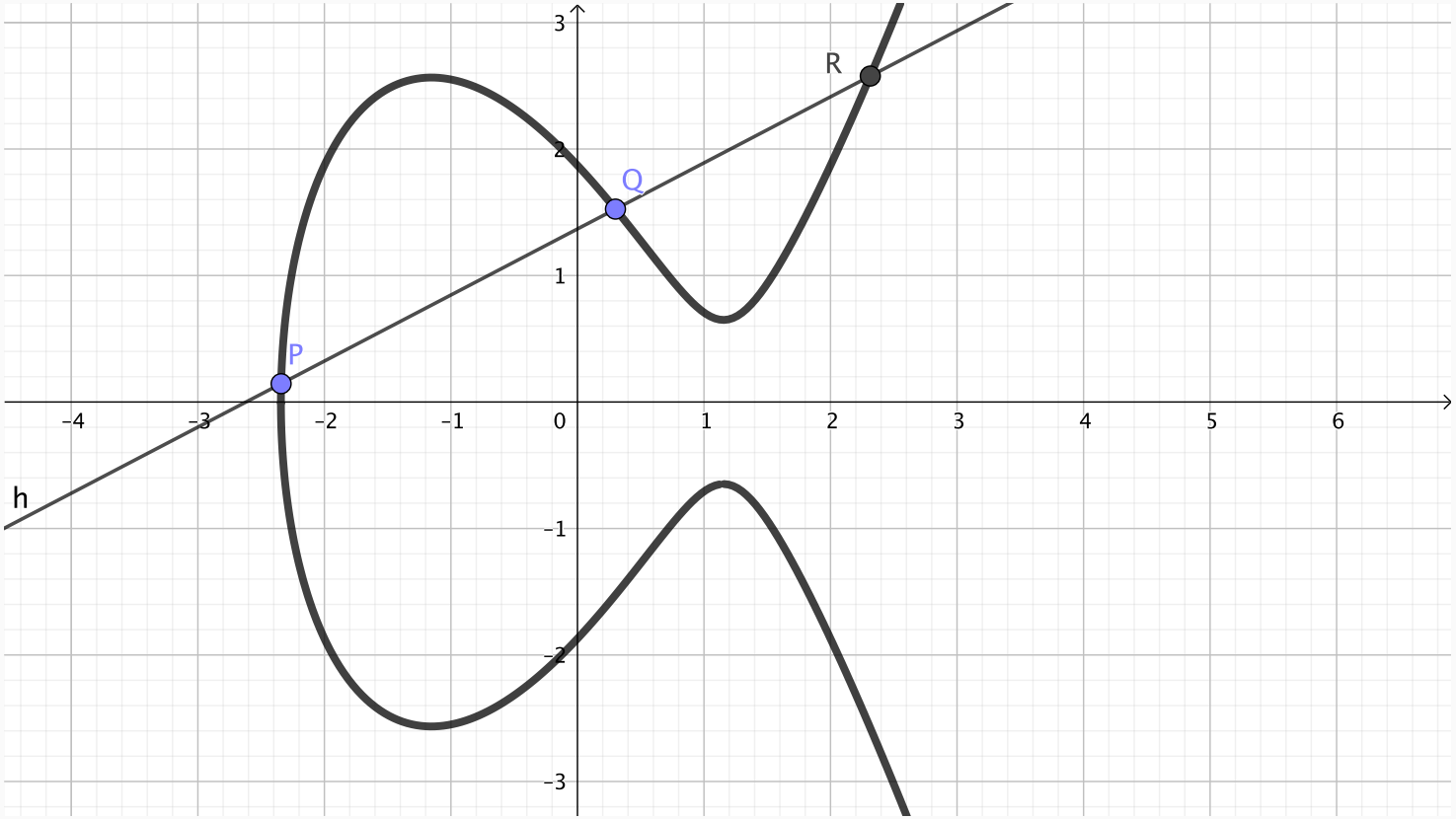
Sum of points

1. Pick a pair of points



Sum of points

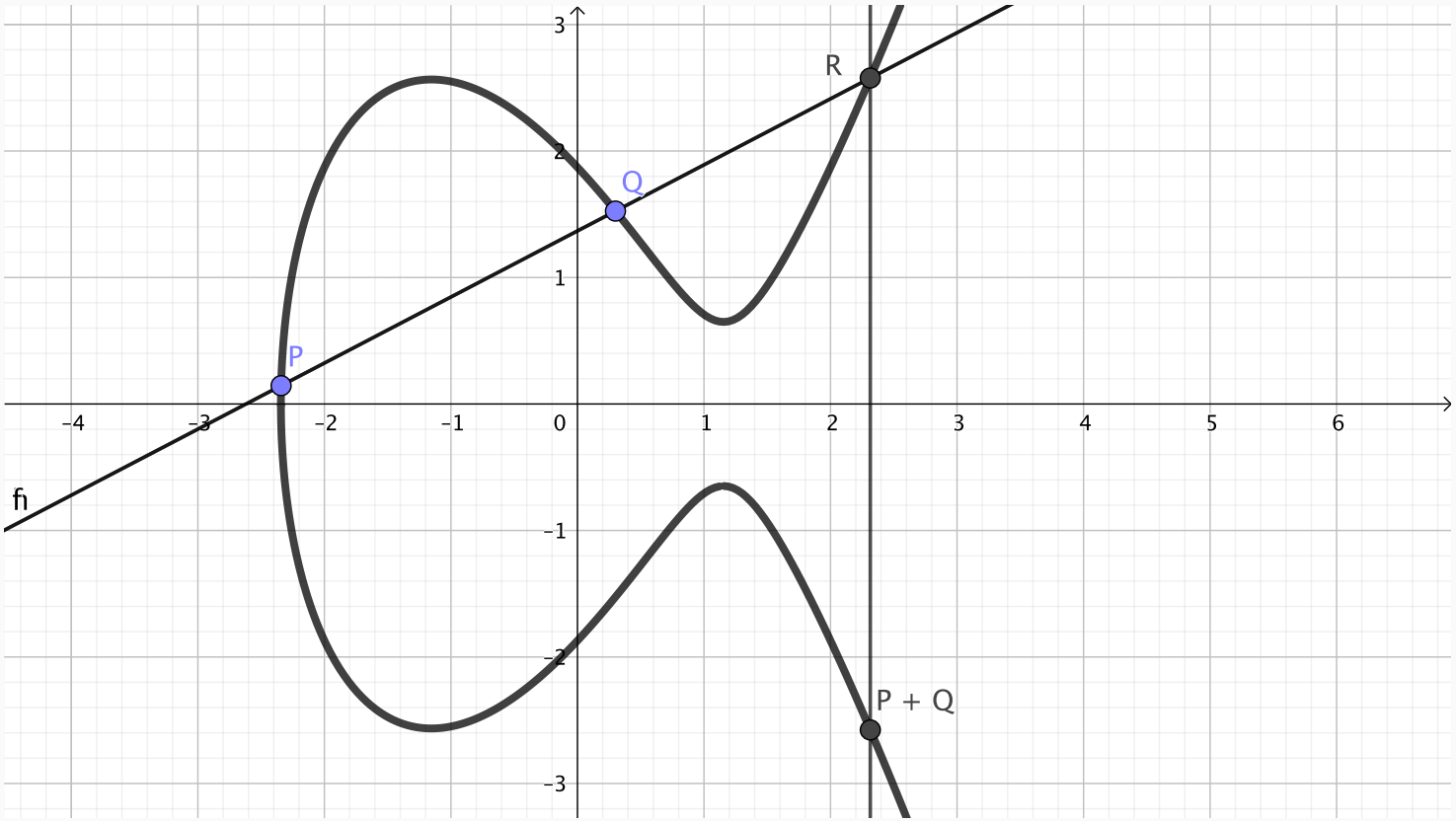
2. Draw a straight line through them



(it intersects the curve again!)

Sum of points

3. Reflect the third point R vertically to get $P + Q$

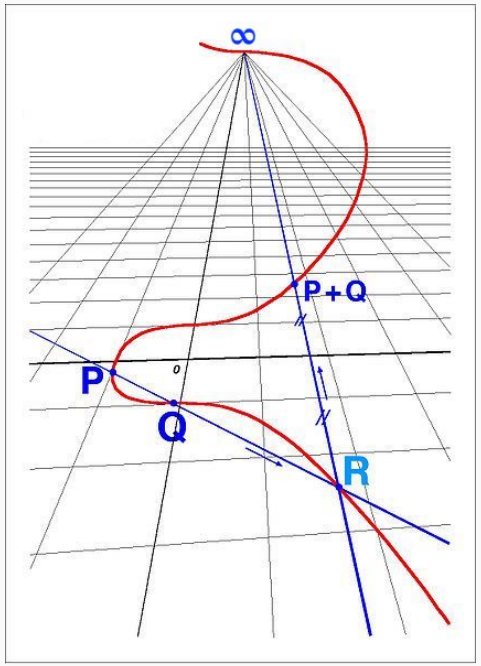


The Point Infinity

We also consider a point at infinity, under the rule

$$P + \infty = P$$

It is visualized like so

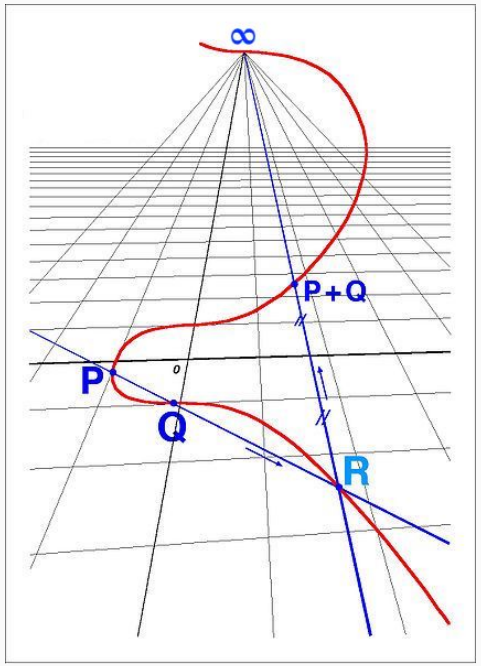


The Point Infinity

We also consider a point at infinity, under the rule

$$P + \infty = P$$

It is visualized like so



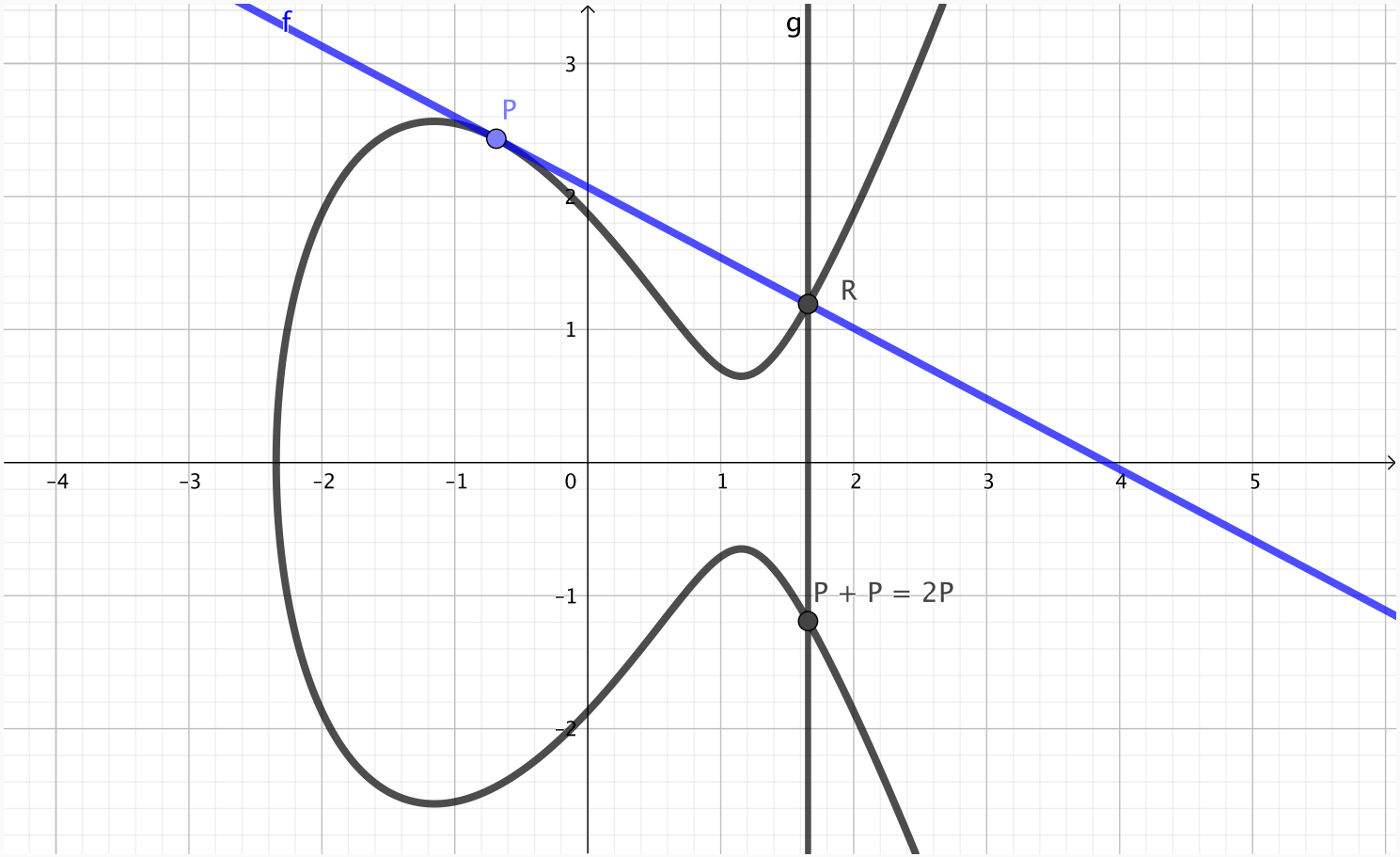
This means we can subtract points! $(P + Q) + R = \infty$, and so $R = -(P + Q)$.

Sum of points

What about adding $P + P$?

Sum of points

What about adding P + P? Draw the tangent!



Addition formulas (not that important!)

To add $P = (x_1, y_1)$ and $Q = (x_2, y_2)$,

- If $x_1 \neq x_2$, then

$$x_3 = m^2 - x_1 - x_2,\ y_3 = m(x_1 - x_3) - y_1,$$

$$\text{where } m = \frac{y_2 - y_1}{x_2 - x_1}.$$

- If $x_1 = x_2$ and $y_1 \neq y_2$, then $P + Q = \infty$.
- If $x_1 = x_2$ and $y_1 \neq 0$, then

$$x_3 = m^2 - 2x_1,\ y_3 = m(x_1 - x_3) - y_1$$

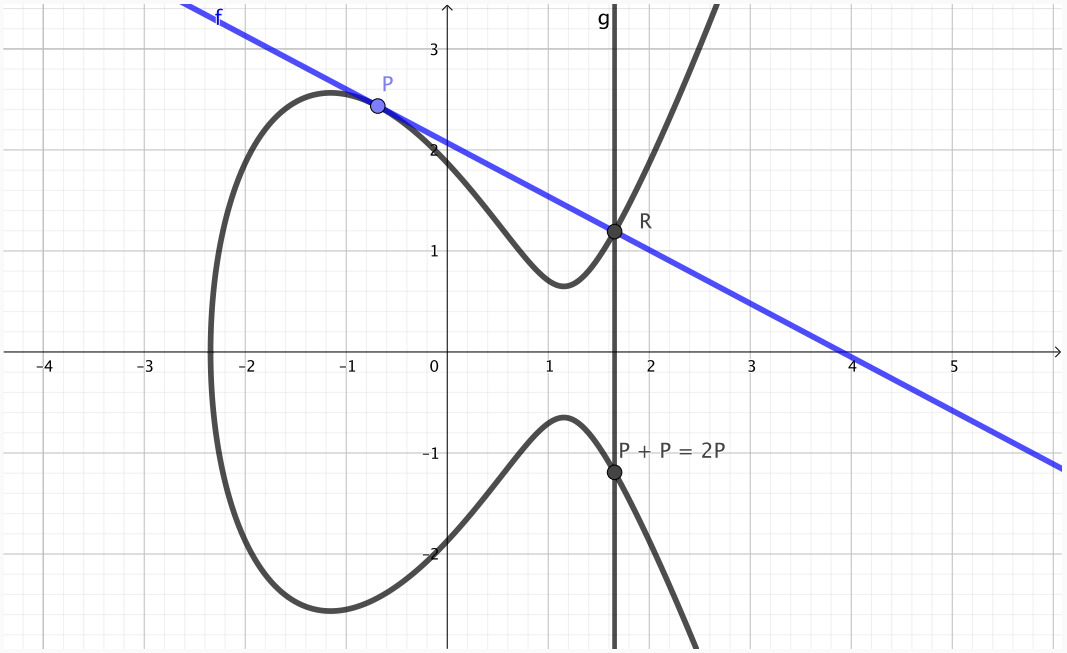
$$\text{where } m = \frac{3x_1^2 + A}{2y_1}.$$

- If $x_1 = x_2$ and $y_1 = y_2 = 0$, then $P + Q = \infty$.
- $P + \infty = P$.

Multiplication of points

If n is an integer and P is a point, we may define $\mathbf{n} \times \mathbf{P}$ as adding P to itself n times.

(We just saw an example of what $2P$ looks like)



Curves *mod p*

Consider a prime number p.

[3] Sometimes called $GF(p)$, the Galois Field of p elements, and also $\mathbb{Z}/p\mathbb{Z}$.

Curves *mod p*

Consider a prime number p. We can consider our numbers (point coordinates and constants) on $\mathbb{F}_p$, the Finite Field with p elements.[3] In practice, **we will be working mod p**, or % p.

[3] Sometimes called $GF(p)$, the Galois Field of p elements, and also $\mathbb{Z}/p\mathbb{Z}$.

Curves *mod p*

Consider a prime number p. We can consider our numbers (point coordinates and constants) on $\mathbb{F}_p$, the Finite Field with p elements.[3] In practice, **we will be working mod p**, or % p. For instance, if our curve is

$$y^2 = x^3 + 1242617x + 21985,$$

then *mod* 7 it becomes

$$y^2 = x^3 + 5x + 5,$$

[3] Sometimes called $GF(p)$, the Galois Field of p elements, and also $\mathbb{Z}/p\mathbb{Z}$.

Curves *mod p*

Consider a prime number p. We can consider our numbers (point coordinates and constants) on $\mathbb{F}_p$, the Finite Field with p elements.[3] In practice, **we will be working mod p**, or % p. For instance, if our curve is

$$y^2 = x^3 + 1242617x + 21985,$$

then *mod* 7 it becomes

$$y^2 = x^3 + 5x + 5,$$

and now the **only** points on the curve are

$$\{(1,2),(1,5),(2,3),(2,4),(5,1),(5,6)\}$$

[3] Sometimes called $GF(p)$, the Galois Field of p elements, and also $\mathbb{Z}/p\mathbb{Z}$.

Curves mod p

How do you find all points?

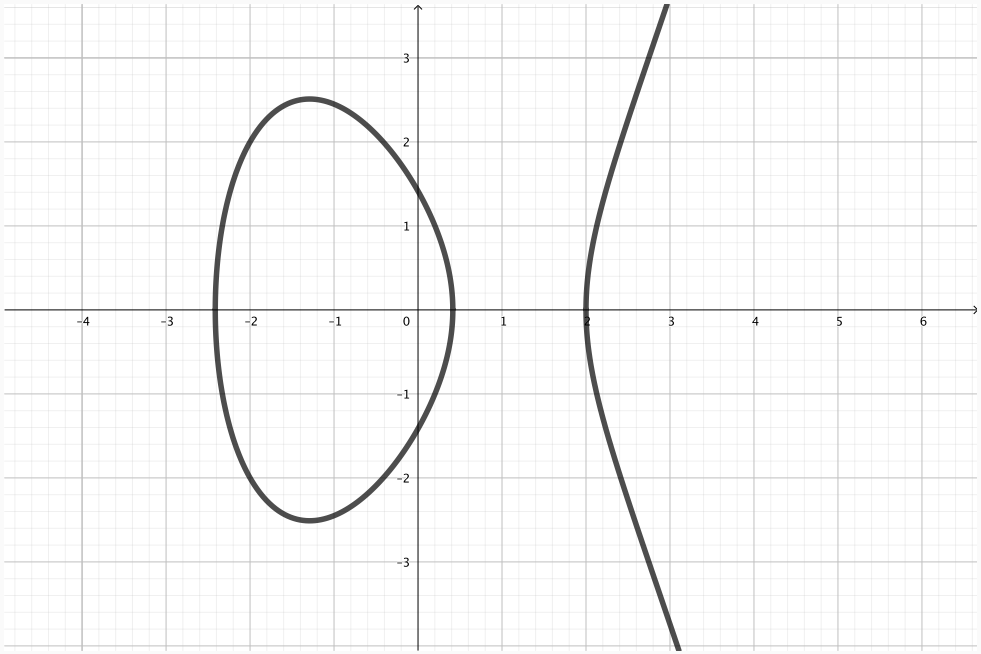
In [1]:
```
print(
    [(x,y)
         for x in range(0,7)
         for y in range(0,7)

     if (y**2 - x**3 - 5*x - 5) % 7 == 0]
)
```

```
[(1, 2), (1, 5), (2, 3), (2, 4), (5, 1), (5, 6)]
```

Curves mod p

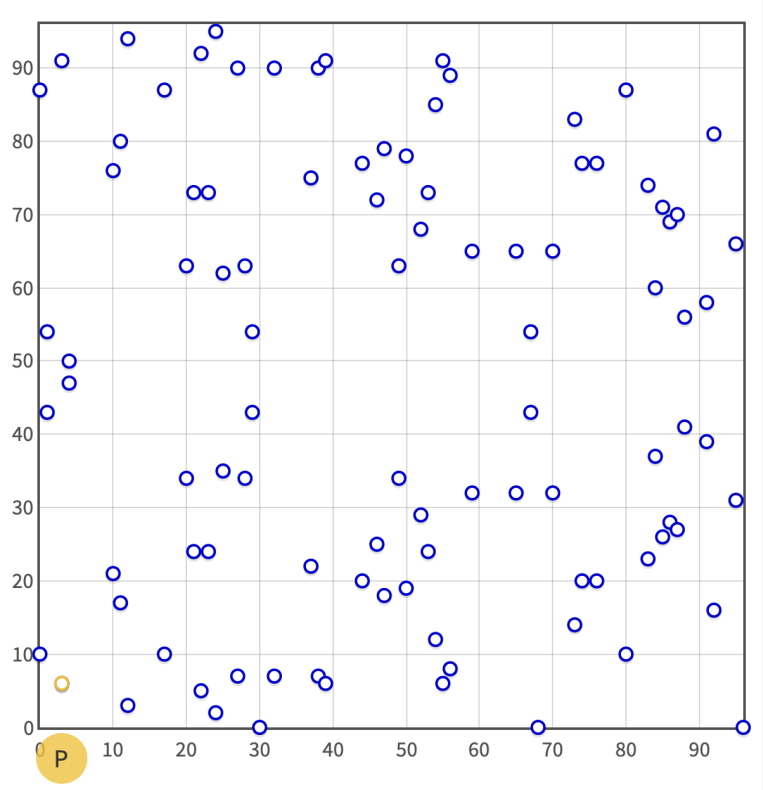
There's a catch!

If the only values are $0, 1, 2, \ldots, p-1$, then our curve goes from looking like this...



Curves mod p

...to looking like this:[4]



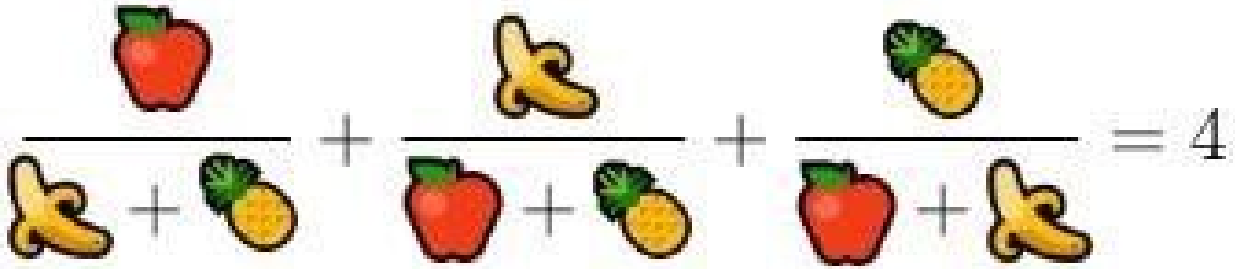
Graph of $y^2 = x^3 + 2x + 3 \mod 97$

[4] https://bit.ly/2GEZoLL

Wow, such big numbers

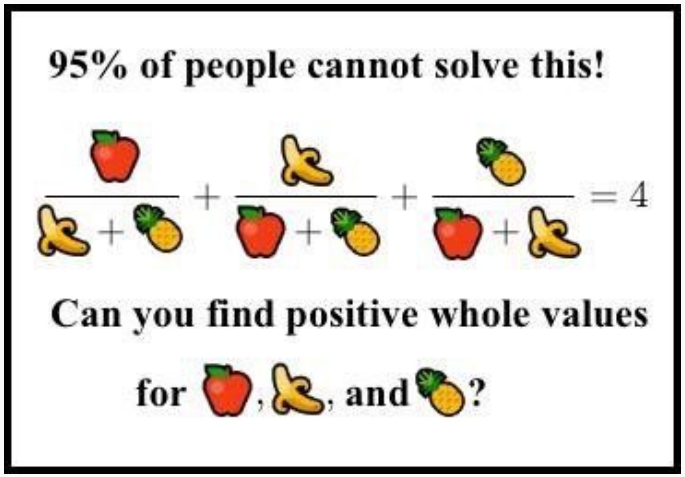
Wow, such big numbers

95% of people cannot solve this!



Can you find positive whole values for , , and ?

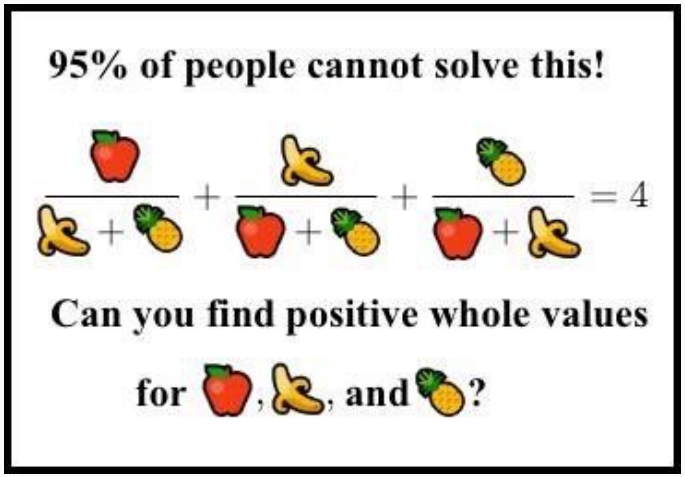
Wow, such big numbers



If you allow for negative values, then you can take:

$$a = 4,\ b = -1,\ c = 11$$

Wow, such big numbers



The smallest such values are:[5]

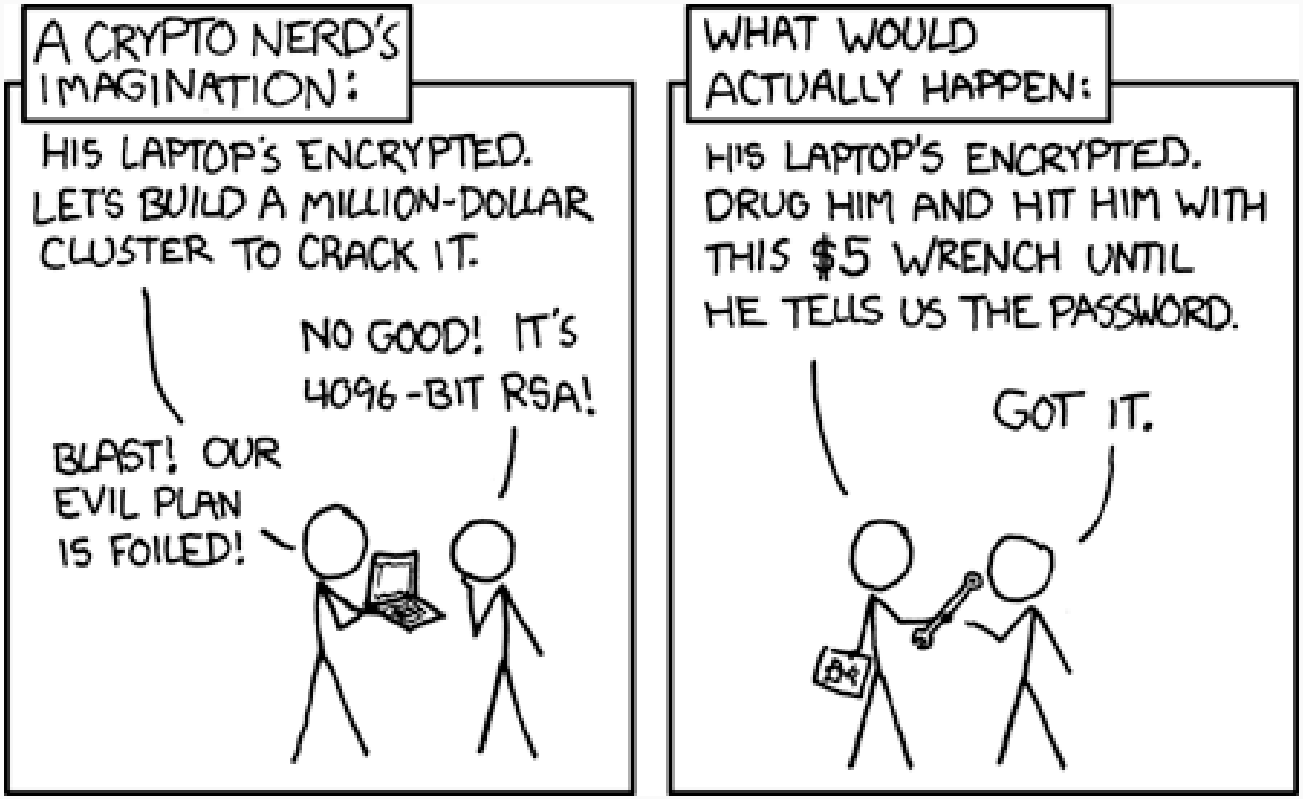
$a = 154476802108746166441951315019919837485664325669565431700026634898253202035277999$

$b = 368751317941299998271978115652254748254929799689719709962831374716372246340555 79$

$c = 437361267792869725786125260237139015281653755816161361862143799337842346777 2036$

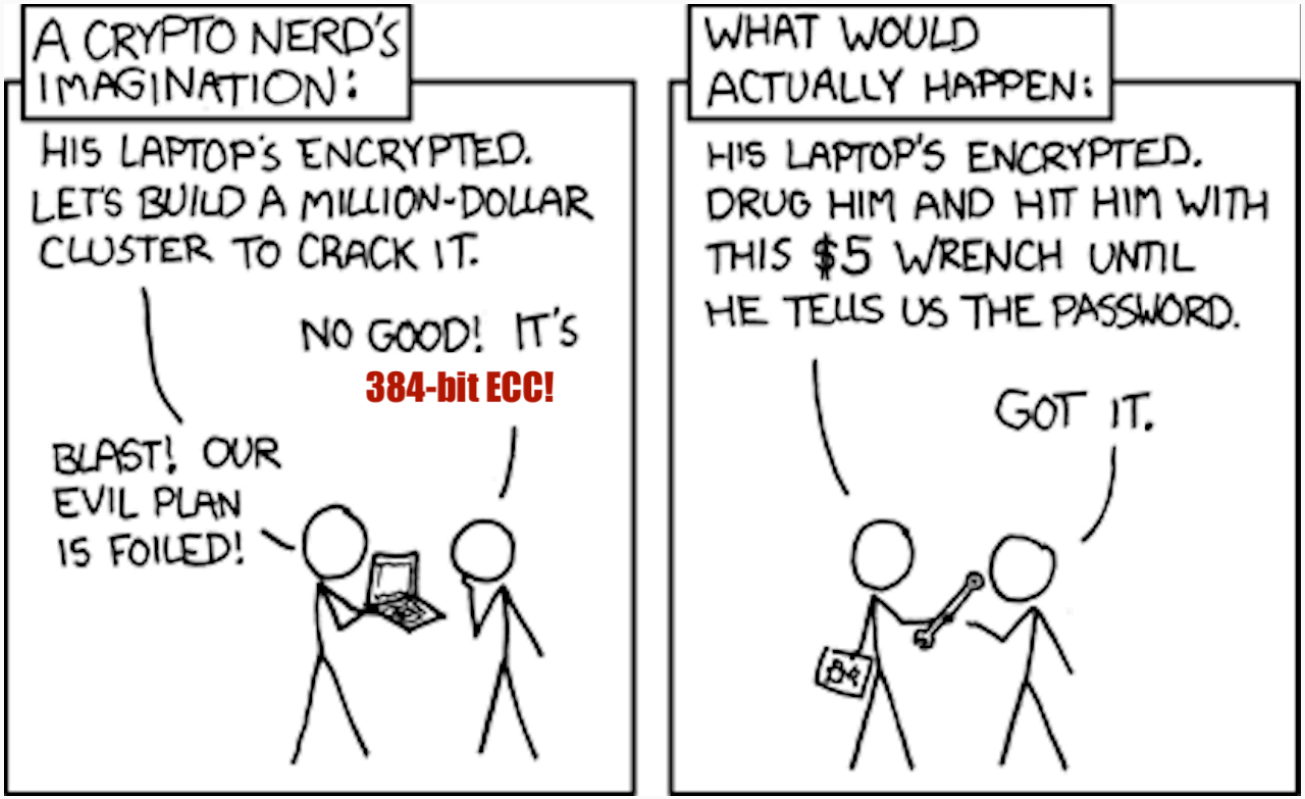
[5] https://qr.ae/TW7qSM

Elliptic Curve Cryptography



https://xkcd.com/538/

Elliptic Curve Cryptography



https://xkcd.com/538/

Elliptic Curve Cryptography Scheme

The typical scheme for ECC consists of:

Elliptic Curve Cryptography Scheme

The typical scheme for ECC consists of:

- The parameters of the curve equation ($y^2 = x^3 + \mathbf{A}x + \mathbf{B}$, $x^2 + y^2 = 1 - \mathbf{d}x^2y^2$...),
- A prime number p,

Elliptic Curve Cryptography Scheme

The typical scheme for ECC consists of:

- The parameters of the curve equation ($y^2 = x^3 + \mathbf{A}x + \mathbf{B}$, $x^2 + y^2 = 1 - \mathbf{d}x^2y^2$...),
- A prime number p,
- A point $P = (x_0, y_0)$ on the curve, called a generator,

Elliptic Curve Cryptography Scheme

The typical scheme for ECC consists of:

- The parameters of the curve equation ($y^2 = x^3 + \mathbf{A}x + \mathbf{B}$, $x^2 + y^2 = 1 - \mathbf{d}x^2y^2$...),
- A prime number p,
- A point $P = (x_0, y_0)$ on the curve, called a generator,
- A number n for which $nP = \infty$, and for all $0 < k < n$, $kP \neq \infty$ (the order of P).

Elliptic Curve Cryptography Scheme

The typical scheme for ECC consists of:

- The parameters of the curve equation ($y^2 = x^3 + \mathbf{A}x + \mathbf{B}$, $x^2 + y^2 = 1 - \mathbf{d}x^2y^2$...),
- A prime number p,
- A point $P = (x_0, y_0)$ on the curve, called a generator,
- A number n for which $nP = \infty$, and for all $0 < k < n$, $kP \neq \infty$ (the order of P).

We want n to be large, and $\gcd(p, n) = 1$.

The first two points are referenced as *CURVE*, and the scheme as $(CURVE, n, P)$.

Elliptic Curve Diffie Hellman

We now have the curve parameters:

$$\begin{cases} y^2 = x^3 + Ax + B \\ p, P = (x_0, y_0), n \end{cases}$$

In this scheme, a private key is an integer d_A between 1 and $n - 1$.

The corresponding public key is $d_A P$.

Key Exchange // ECDH

ALICE	BOB
$d_A, d_A P$	$d_B, d_B P$
(Alice gets $d_B P$)	(Bob gets $d_A P$)
$d_A \cdot (d_B P)$	$d_B \cdot (d_A P)$

They both have **the same point $\mathbf{d_A d_B P} = (\mathbf{x}, \mathbf{y})$.**

Take x, or its hash, as the shared symmetric key.

ECDSA

We can also perform **Digital Signature** using a private key and a *random integer* (used only once).

ECDSA

We can also perform **Digital Signature** using a private key and a *random integer* (used only once).

Let h be the hashed message, and choose a random integer r. Then compute

$$s = r^{-1}(h + d_A x),$$

with $x = x(rP)$. The signature is

$$(h, s, rP).$$

ECDSA - Verification

The curve is $(CURVE, n, P)$ Alice's public key is $Q = d_A P$, and the signature is (h, s, rP). To verify it, compute:

$$u_1 = s^{-1}h \mod n$$

$$u_2 = s^{-1}x(rP) \mod n$$

$$\mathbf{V := u_1 P + u_2 Q}.$$

If the message is correct, then the verification holds:

$$V = u_1 P + u_2 Q = s^{-1}hP + s^{-1}xd_A P = s^{-1}(h + xd_A)P = rP.$$

Discrete Logarithm Problem

Say you encounter a public key Q and a curve $(CURVE, n, P)$, we want to find k such that

$$Q = kP$$

k will be the private key!

Discrete Logarithm Problem

Say you encounter a public key Q and a curve $(CURVE, n, P)$, we want to find k such that

$$Q = kP$$

k will be the private key!

This is a Discrete Logarithm Problem:

> *Given P and kP, find k.*

DLP “Attacks”

The are general algorithms to solve a DLP:

- Giant Step-Baby Step (memory intensive)
- Pollard’s Rho and Lambda
- Pohling-Hellman

MOV Attack

Based on pairings, i.e. mapping pairs of points to a finite field $\mathbb{F}_{p^k}$ where the discrete logarithm is easier to compute.

Still expensive in most cases.

Anomalous curves

If an elliptic curve has exactly p points over $\mathbb{F}_p$, we say it is **anomalous**.

We don't use them for cryptography, because they can be attacked in polynomial time.

The p-adic attack

We will attack curves mod p having exactly p points (p a prime, as always).

The p-adic attack

We will attack curves mod p having exactly p points (p a prime, as always).

As always, P is the generator, and Q is a public key. We know there is some k with

$$Q = kP$$

The p-adic attack

We will attack curves mod p having exactly p points (p a prime, as always).

As always, P is the generator, and Q is a public key. We know there is some k with

$$Q = kP$$

It would be nice to just do $k = \frac{Q}{P}$ but...

The p-adic attack

We will attack curves mod p having exactly p points (p a prime, as always).

As always, P is the generator, and Q is a public key. We know there is some k with

$$Q = kP$$

It would be nice to just do $k = \frac{Q}{P}$ but... We can't really divide points!

The p-adic attack (background)

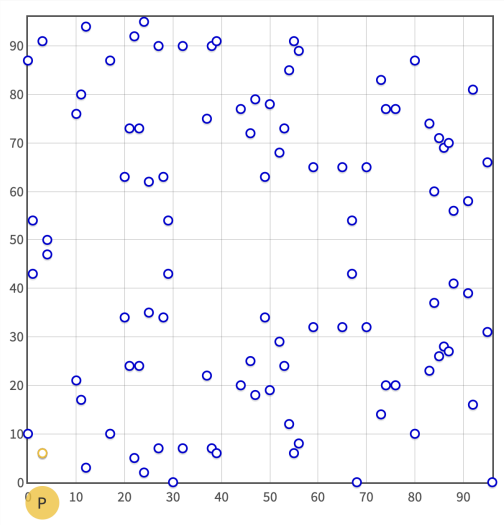
If we have a curve $E: \ y^2 = x^3 + Ax + B$ in $\mathbb{F}_p$ and two points P and Q, we can *lift* it to a curve $\tilde{E}: \ y^2 = x^3 + \tilde{A}x + \tilde{B}$ and two points $\tilde{P}$ and $\tilde{Q}$ with

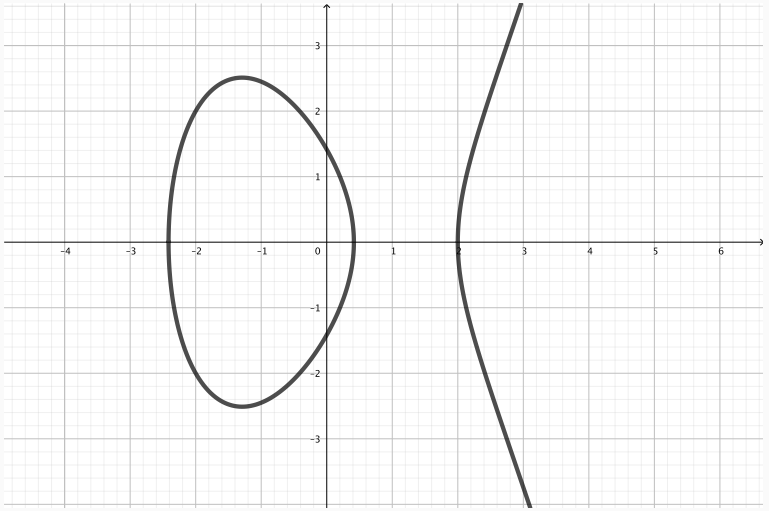
$$\begin{cases} A \equiv \tilde{A}, \ B \equiv \tilde{A} \mod p \\ P \equiv \tilde{P}, \ Q \equiv \tilde{Q} \mod p \end{cases}.$$

The p-adic attack (background)

If we have a curve $E: \ y^2 = x^3 + Ax + B$ in $\mathbb{F}_p$ and two points P and Q, we can *lift* it to a curve $\tilde{E}: \ y^2 = x^3 + \tilde{A}x + \tilde{B}$ and two points $\tilde{P}$ and $\tilde{Q}$ with

$$\begin{cases} A \equiv \tilde{A}, \ B \equiv \tilde{A} \mod p \\ P \equiv \tilde{P}, \ Q \equiv \tilde{Q} \mod p \end{cases}.$$





The p-adic attack (background)

If we have a curve E in rationals, we can consider the reduction map modulo p, which takes each coordinate mod p.

The p-adic attack (background)

If we have a curve E in rationals, we can consider the reduction map modulo p, which takes each coordinate mod p.

$$red_p : E(\mathbb{Q}) \to E(\mathbb{F}_p)$$
$$\left(\frac{a}{b}, \frac{c}{d}\right) \mapsto (ab^{-1}, cd^{-1}) \mod p$$
$$\infty \mapsto \infty$$

The p-adic attack (background)

If we have a curve E in rationals, we can consider the reduction map modulo p, which takes each coordinate mod p.

$$red_p : E(\mathbb{Q}) \to E(\mathbb{F}_p)$$
$$\left(\frac{a}{b}, \frac{c}{d}\right) \mapsto (ab^{-1}, cd^{-1}) \mod p$$
$$\infty \mapsto \infty$$

If p divides b or d, then the point goes to ∞.

The p-adic attack (background)

If we have a curve E in rationals, we can consider the reduction map modulo p, which takes each coordinate mod p.

$$red_p : E(\mathbb{Q}) \to E(\mathbb{F}_p)$$
$$\left(\frac{a}{b}, \frac{c}{d}\right) \mapsto (ab^{-1}, cd^{-1}) \mod p$$
$$\infty \mapsto \infty$$

If p divides b or d, then the point goes to ∞.

The map red_p is linear (as in linear algebra), meaning

$$red_p(nP) = n \times red_p(P)$$
$$red_p(P + Q) = red_p(P) + red_p(Q).$$

The p-adic attack (background)

If a curve E has n points in $\mathbb{F}_p$, then all points satisfy:

$$nP = \infty.$$

The p-adic attack (background)

Given a curve $E(\mathbb{Q})$, take the set of points with denominator divisible by p:[6]

$$E_r = \left\{ \left(\frac{a}{b}, \frac{c}{d} \right) \in E(\mathbb{Q}) \mid p^{2r} | b,\ p^{3r} | d \right\}.$$

[6] This is the p-adic bit of the talk!

The p-adic attack (background)

Given a curve $E(\mathbb{Q})$, take the set of points with denominator divisible by p:[6]

$$E_r = \left\{ \left(\frac{a}{b}, \frac{c}{d} \right) \in E(\mathbb{Q}) \mid p^{2r} | b,\ p^{3r} | d \right\}.$$

If P is in E_1, then p divides the denominators of the point, and so $red_p(P) = \infty$ (and viceversa).

[6] This is the p-adic bit of the talk!

The p-adic attack (background)

Given a curve $E(\mathbb{Q})$, take the set of points with denominator divisible by p:[6]

$$E_r = \left\{ \left(\frac{a}{b}, \frac{c}{d} \right) \in E(\mathbb{Q}) \mid p^{2r} | b,\ p^{3r} | d \right\}.$$

If P is in E_1, then p divides the denominators of the point, and so $red_p(P) = \infty$ (and viceversa).

We can consider the **p-adic logarithm**

$$\lambda_1 \colon E_1 \to \mathbb{Z}/p^4\mathbb{Z}$$
$$(x, y) \mapsto \frac{1}{p}\frac{x}{y} \mod p^4$$

[6] This is the p-adic bit of the talk!

The p-adic attack

- Begin with a curve $E(\mathbb{F}_p)$ with exactly p points, P the generator and Q a public key.

The p-adic attack

- Begin with a curve $E(\mathbb{F}_p)$ with exactly p points, P the generator and Q a public key.
- Lift E, P and Q to $\tilde{E}(\mathbb{Q})$, $\tilde{P}$ and $\tilde{Q}$ in rational numbers.

The p-adic attack

- Begin with a curve $E(\mathbb{F}_p)$ with exactly p points, P the generator and Q a public key.
- Lift E, P and Q to $\tilde{E}(\mathbb{Q})$, $\tilde{P}$ and $\tilde{Q}$ in rational numbers.
- Let $\tilde{P}_1 = p \cdot \tilde{P}$, $\tilde{Q}_1 = p \cdot \tilde{Q}$. Then $\tilde{P}_1 \in \tilde{E}_1$, since $red_p(\tilde{P}_1) = red_p(p \cdot \tilde{P}) = p \cdot red_p(\tilde{P}) = \infty$. Same with $\tilde{Q}_1$.

The p-adic attack

- Begin with a curve $E(\mathbb{F}_p)$ with exactly p points, P the generator and Q a public key.
- Lift E, P and Q to $\tilde{E}(\mathbb{Q})$, $\tilde{P}$ and $\tilde{Q}$ in rational numbers.
- Let $\tilde{P}_1 = p \cdot \tilde{P}$, $\tilde{Q}_1 = p \cdot \tilde{Q}$. Then $\tilde{P}_1 \in \tilde{E}_1$, since $red_p(\tilde{P}_1) = red_p(p \cdot \tilde{P}) = p \cdot red_p(\tilde{P}) = \infty$. Same with $\tilde{Q}_1$.
- If $\tilde{P}_1 \notin \tilde{E}_2$, then

$$k \equiv \frac{\lambda_1(\tilde{Q}_1)}{\lambda_1(\tilde{P}_1)} \mod p$$

The p-adic attack (Proof)

We can easily check that k is the **private key**, i.e. $kP = Q$:

$$k\lambda_1(\tilde{P}_1) - \lambda_1(\tilde{Q}_1) = \lambda_1(k\tilde{P}_1 - \tilde{Q}_1) = \lambda_1(kp\tilde{P} - p\tilde{Q}) = p\lambda_1(k\tilde{P} - \tilde{Q}) \equiv 0$$

and we can do this because

$$\infty = kP - Q = red_p(k\tilde{P} - \tilde{Q}) \implies k\tilde{P} - \tilde{Q} \in \tilde{E}_1.$$

The p-adic attack (Proof)

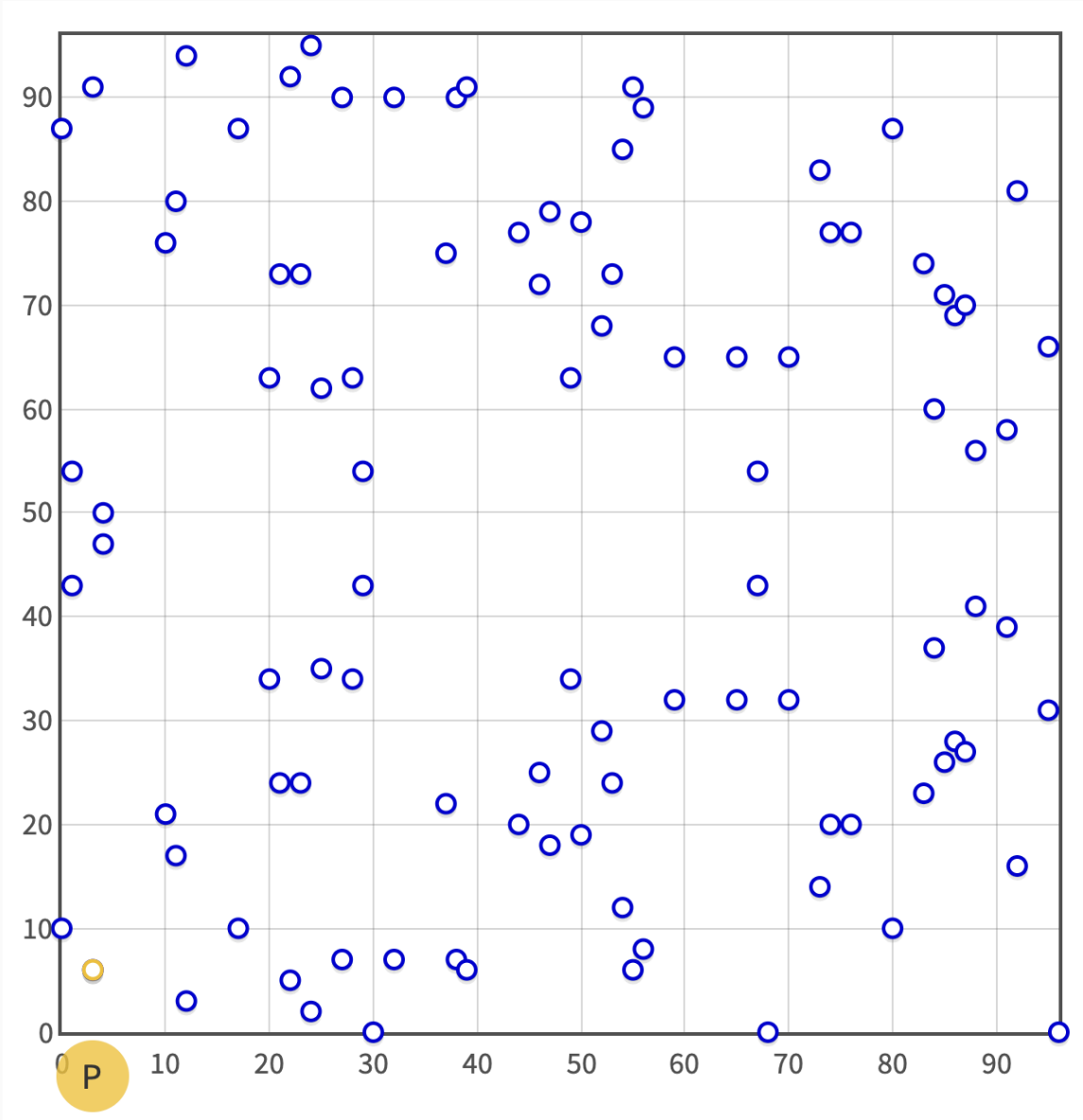
We can easily check that k is the **private key**, i.e. $kP = Q$:

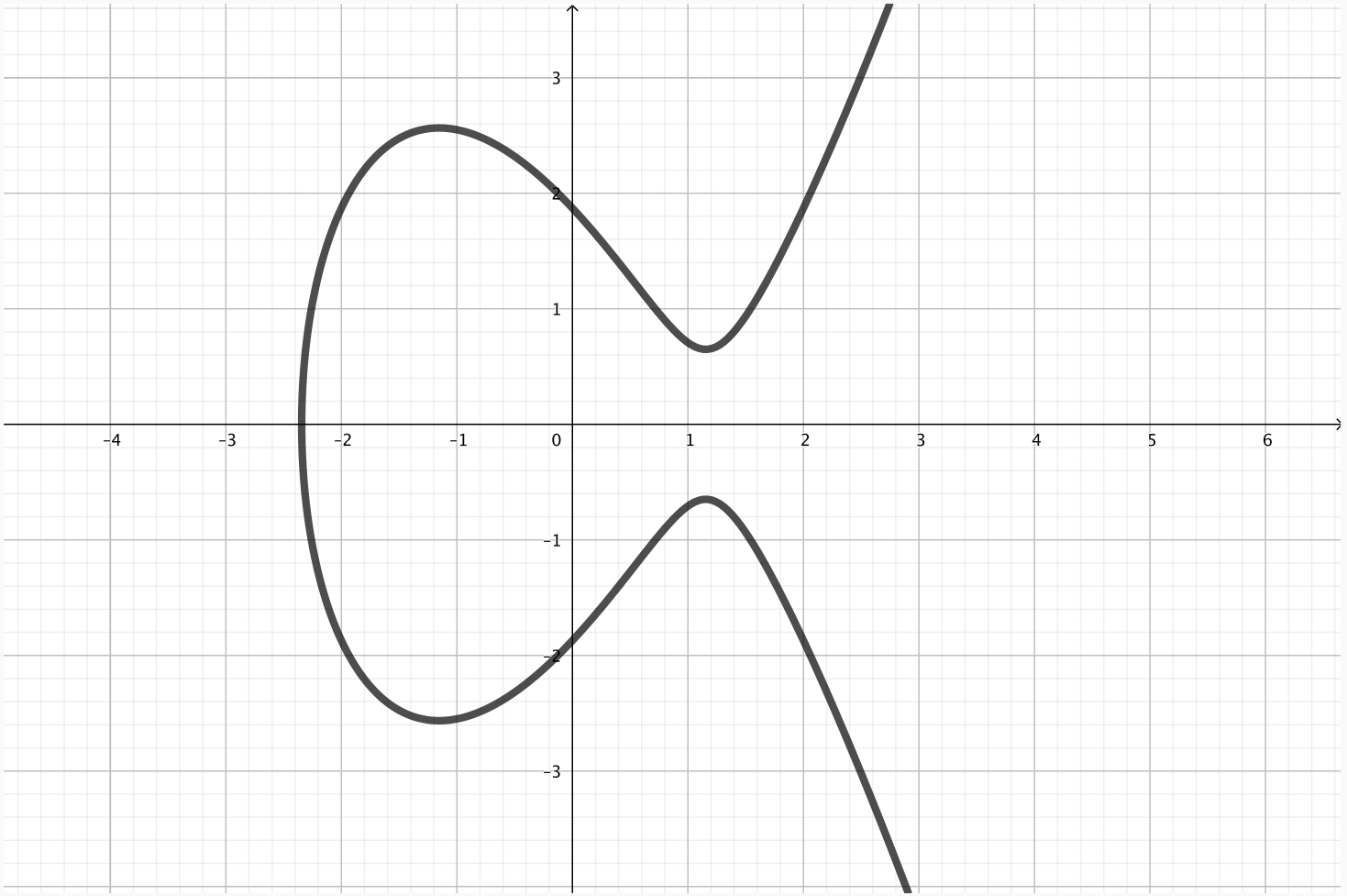
$$k\lambda_1(\tilde{P}_1) - \lambda_1(\tilde{Q}_1) = \lambda_1(k\tilde{P}_1 - \tilde{Q}_1) = \lambda_1(kp\tilde{P} - p\tilde{Q}) = p\lambda_1(k\tilde{P} - \tilde{Q}) \equiv 0$$

and we can do this because

$$\infty = kP - Q = red_p(k\tilde{P} - \tilde{Q}) \implies k\tilde{P} - \tilde{Q} \in \tilde{E}_1.$$

The numbers involved in the computation of k are way too big, but we can work modulo p^2 (see references).


90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
P


3
2
1
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
4
5
6

Some references

- https://www.crypto101.io
- Introduction to ECC https://bit.ly/2GV19pR
- Video introduction to ECC at CCC https://bit.ly/2nEmACI
- Smart's attack on anomalous curves https://www.hpl.hp.com/techreports/97/HPL-97-128.pdf
- https://safecurves.cr.yp.to
- Elliptic Curves: Number theory and cryptography (L. Washington).

To sum up

- Don’t fear elliptic curves! Advanced stuff can look difficult, but the basic theory is attainable.
- They are more lightweight than other public key systems.
- Theoretical attacks can appear based on really strange properties.
- **If you need to implement some ECC, go read:**

  `https://safecurves.cr.yp.to`

  and its references.