# Elliptic curves, number theory and cryptography

## Week 1, Lecture 1

Aurore Guillevic

Aarhus University

Spring semester, 2022

These slides at

`https://members.loria.fr/AGuillevic/files/Enseignements/AU/lectures/lecture01.pdf`

# Outline

Administrative info

Introduction

Addition Law

Main results

Recap on finite fields

Recap on complexity

# Course materials

- Page in the catalog: `https://www.kursuskatalog.au.dk/en/course/112277/Elliptic-Curves-Number-Theory-and-Cryptography`
- Brightspace: `https://brightspace.au.dk/d2l/home/55068`
- Calendar: `https://timetable.au.dk/schedule`
    - Add timetable → Module → search for `Elliptiske kurver`
    - Full name is `Elliptiske kurver - talteori og kryptografi F22 Aarhus - 550122U015`
    - Starts Week 5 (January 31)

# Course materials

Book: **Elliptic curves, number theory and cryptography**, Lawrence C. Washington

SageMath library: a mathematical software suite based on Python, open-source.

Additional references:

- Steven Galbraith's book *Mathematics of public key cryptography* is freely available at `https://www.math.auckland.ac.nz/~sgal018/crypto-book/main.pdf`
- Joseph H. Silverman, John Tate, *Rational points on elliptic curves* `https://link.springer.com/book/10.1007/978-3-319-18588-0` also available in PDF at `https://www.kb.dk/en`

# SageMath installation

Download at `https://www.sagemath.org/download.html` There is a *mirror* at `https://mirror.dogado.de/sage/index.html`

## Windows:

Download `SageMath-9.3-Installer-v0.6.3.exe` (820.72MB) from either the mirror `https://mirror.dogado.de/sage/win/index.html` or the main server `https://github.com/sagemath/sage-windows/releases`

## MacOS:

What does *About this Mac* says?

- Intel: Download `sage-9.4-OSX_11.2.3-x86_64.tar.bz2` from `https://mirror.dogado.de/sage/osx/intel/index.html`
- PowerPC: Download one of the files from `https://mirror.dogado.de/sage/osx/powerpc/index.html`

## Linux:

Choose the file according to your architecture and Linux distribution and version at `https://mirror.dogado.de/sage/linux/index.html`

# Course Schedule

Lectures on Tuesdays 08:00 – 10:00, building 1532 room 314
Tutorials on Thursdays, 15:00 – 17:00, building 1532 room 314
Guidance (office hours) on Tuesdays, 13:00 – 14:00 at the CS building (Nygaard 1553) 3rd floor, room 395 (or 387)

From Week 5 (January 31) to Week 14 (April 7)
Break on week 15 (April 11 & April 14)
Then from Week 16 (April 18) to Week 20 (May 19)

By-weekly handins are mandatory to take the final exam

Oral exam in June

# The instructors

Aurore and Diego are cryptographers at the department of Computer Science

Diego F. Aranha, associate professor
`dfaranha@cs.au.dk`

Aurore Guillevic, visiting researcher from France
`aurore.guillevic@inria.fr`

Survey

- who is from Maths?
- who is from Computer Science?
- who took Ivan Damgård's course on Cryptography?
- who has a laptop? Windows? Mac? Linux?
- who already succeeded in installing SageMath?

# Content of this course

- Elliptic curves over a field $K$, from group law to pairings
- Elliptic curves over finite fields: Frobenius, point counting, supersingular curves
- Elliptic curves in cryptography
- Other number-theoretic hard problems in crypto:
  integer factorization, discrete logarithm computation
- pairings on elliptic curves for crypto
- Guest lecture: elliptic curves over binary fields
- Elliptic curves over $\mathbb{Q}$
- Guest lecture: pairing-based cryptography
- hot topic in cryptography: isogenies, post-quantum crypto

# Outline

Administrative info

Introduction

Addition Law

Main results

Recap on finite fields

Recap on complexity

# Elliptic curves in cryptography

- 1985 (published in 1987) Hendrik Lenstra Jr., Elliptic Curve Method (ECM) for integer factoring
- 1985, Koblitz, Miller: Elliptic Curves over a finite field form a group suitable for Diffie–Hellman key exchange
- 1985, Certicom: company owning patents on ECC
- 2000 Elliptic curves in IEEE P1363 standard
- 2000 Bilinear pairings over elliptic curves
- NSA cipher suite B, elliptic curves for public-key crypto
- 2014: Quasi-polynomial-time algorithm
  for discrete log computation in GF($2^n$), GF($3^m$)
  No more pairings on elliptic curves over these fields
- 2015: Tower Number Field Sieve in GF($p^n$)
  Pairing-friendly curves should have larger key sizes
- 2016: NIST Post-Quantum competition
  Isogenies on elliptic curves

# Widely deployed elliptic curves in cryptosystems

- elliptic curve over the prime field $2^{255} - 19$ of order $8r$ where $r$ is prime
  - `Curve25519` in Montgomery form $E\colon y^2 = x^3 + 48662x^2 + x$
  - `Ed25519` in twisted Edwards form $E\colon -x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$
- NIST P-xxx curves
- . . .

Usage:

- Digital signatures (ECDSA): Play Station, EU Covid Certificate...
- Diffie–Hellman key exchange: open-ssl, TLS...
- Encryption: PGP, ...

# Why elliptic curves?

## Diophantine equations

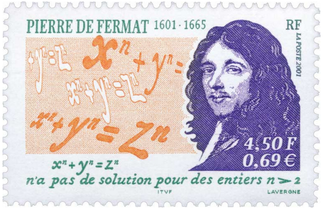
From Diophantus of Alexandria, mathematician
Finding integer or rational solutions to polynomial equations

## Bachet equation $y^2 - x^3 = c$

given an integer $c$, find a cube $x^3$ and a square $y^2$ whose difference is $c$
Claude-Gaspard Bachet de Méziriac (1581–1638)
Translated Diophantus' *Arithmetica* from Greek to latin.

## Fermat's conjecture, a.k.a. Fermat's Last Theorem

Pierre de Fermat (1601–1665)
For $n \geq 3$, the equation $X^n + Y^n = Z^n$ has no solutions in non-zero integers $X, Y, Z$.
Actually not proven by Fermat



https://www.wikitimbres.fr/

# Bachet's equation $y^2 - x^3 = c$

Bachet discovered in 1621 this

duplication formula

If $(x, y)$ is a rational solution, then

$$\left( \frac{x^4 - 8cx}{4y^2}, \frac{-x^6 - 20cx^3 + 8c^2}{8y^3} \right)$$

is another solution in rational numbers.

If $xy \neq 0$ and $c \neq 1, -432$, it gives infinitely many distinct solutions.

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

Start from $5^2 - 3^3 = 25 - 27 = -2$. One obtains

$$(3, 5), \left( \frac{129}{100}, \frac{383}{1000} \right), \left( \frac{2340922881}{58675600}, \frac{113259286337279}{449455096000} \right)$$

In the 1st tutorial we will program in Python this replication formula.

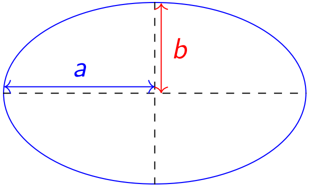
# Example in Washington's book

### Volume and surface

Rearrange a pyramid of height $x$ layers of fruits into a flat square: solve $y^2 = x(x+1)(2x+1)/6$ with integer solutions

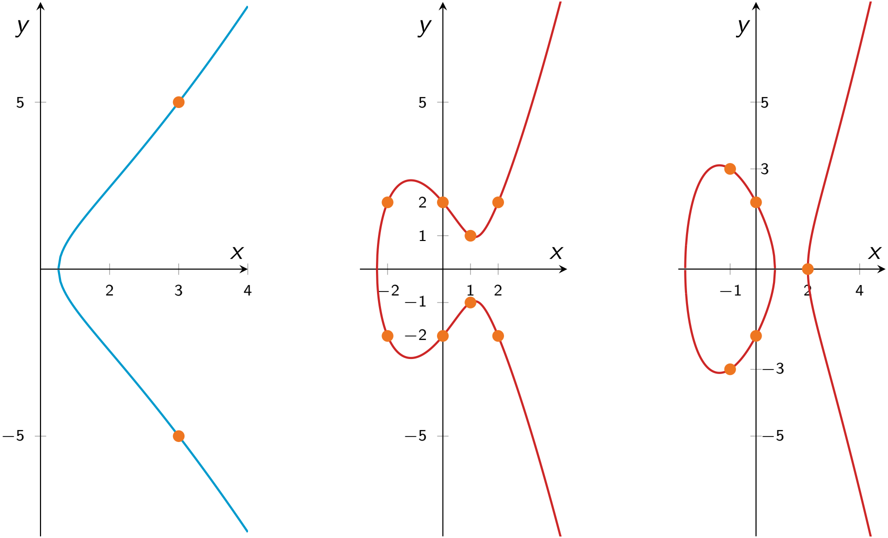
# Conic sections

Ellipses are conic sections defined by $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$



Ellipses are not elliptic curves.
This ellipse has **area** $\pi ab$. What is the **circumference**? → complicated formula with *elliptic integral*.

# Bachet's equation is an elliptic curve



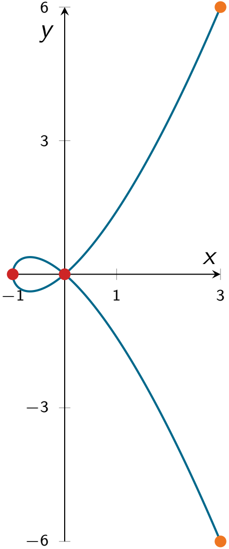
$y^2 - x^3 = -2$

$y^2 = x^3 - 4x + 4$

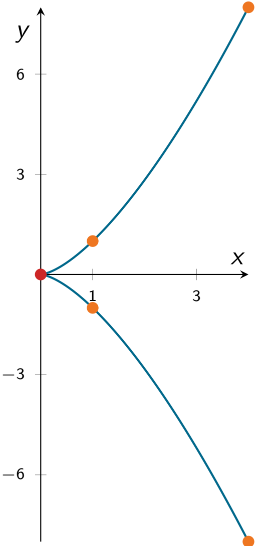
$y^2 = x^3 - 6x + 4$

# Curves with singularities are not elliptic curves

$y^2 = x^2(x+1)$



$y^2 = x^3$



# The curve is smooth

Let $E\colon f(x,y)=0$ over a field $K$.
There is no singular point $(x_0, y_0)$ such that

$$\begin{cases} f(x_0,y_0)=0 \\[2ex] \dfrac{\partial f}{\partial x}(x_0,y_0)=0 \\[2ex] \dfrac{\partial f}{\partial y}(x_0,y_0)=0 \end{cases}$$

where $\partial f/\partial x$, $\partial f/\partial y$ are the partial derivatives.

# Definitions

### Elliptic Curve

An **Elliptic Curve** over a field $K$ is a smooth curve of *genus 1* with a $K$-rational point.

### Genus 1

A curve given by an equation

$$y^2 = f(x), \text{ where } \deg f \in \{3, 4\}$$

has genus 1.

### Structure of Group

Given two points $P(x, y)$, $Q(x', y')$, one can *add* two points $P + Q$ and double a point $P + P$ (algebraic point of view) ans the group law has a geometric meaning.

# Outline

Administrative info

Introduction

Addition Law

Main results

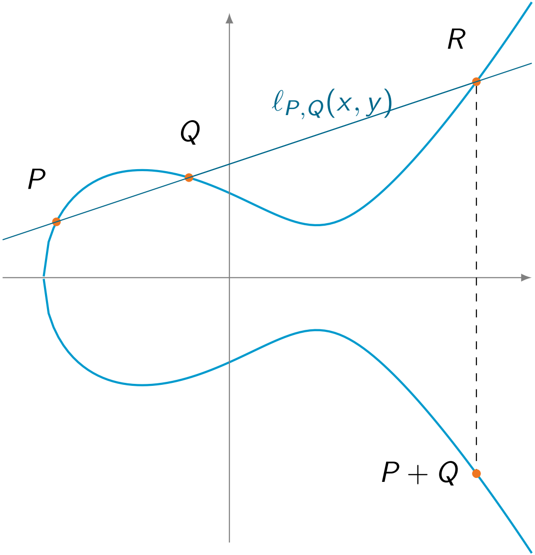
Recap on finite fields

Recap on complexity

# Weierstrass model

- An elliptic curve over a field $K$ of characteristic $\neq 2, 3$ is given by an equation of the form
$$E\colon y^2 = x^3 + ax + b, \text{ with } a, b \in K$$
and $\Delta = -16(4a^3 + 27b^2) \neq 0$ so that $E$ is smooth
(the cubic $x^3 + ax + b$ has simple roots)
- The set of $K$-rational points of an elliptic curve is
$$E(K) = \Big\{(x, y) \in K \times K;\ y^2 = x^3 + ax + b\Big\} \cup \{\mathcal{O}\}$$
- In the general case, one considers the long **Weierstrass** form
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$
where $a_1, a_2, a_3, a_4, a_6 \in K$.

# Chord and tangent rule



$P(x_1, y_1)$, $Q(x_2, y_2)$, $x_1 \neq x_2$

slope $\lambda = \dfrac{\Delta y}{\Delta x} = \dfrac{y_2 - y_1}{x_2 - x_1}$

line $L$ through $P$ and $Q$ has equation

$L\colon y = \lambda(x - x_1) + y_1$

$\rightarrow$ check that $(x_1, y_1) \in L$, $(x_2, y_2) \in L$

compute $\boldsymbol{L} \cap \boldsymbol{E}$ $(x, y) \in L$ and $\in E \Rightarrow$

$$\begin{cases} L\colon y = \lambda(x - x_1) + y_1 \\ E\colon y^2 = x^3 + ax + b \end{cases} \Rightarrow$$

$\big(\lambda(x - x_1) + y_1\big)^2 = x^3 + ax + b$

Solve with SageMath to avoid mistakes

```
# define a polynomial ring for the variables
QQx.<a,b,l,x_1,y_1,x_2,y_2> = QQ[]
QQE.<X,Y> = QQx[]
L = Y - (l*(X-x_1) + y_1); Lx = Y - L
E = Y^2 - X^3 - a*X - b
Eq = E(Y=Lx); Eq # evaluate E at Y = Lx
```

```
# define a polynomial ring for the variables
QQx.<a,b,l,x_1,y_1,x_2,y_2> = QQ[]
QQE.<X,Y> = QQx[]
L = Y - (l*(X-x_1) + y_1); Lx = Y - L
E = Y^2 - X^3 - a*X - b
Eq = E(Y=Lx); Eq # evaluate E at Y = Lx
```

$$-X^3 + \lambda^2 X^2 + (-2x_1\lambda^2 + 2y_1\lambda - a)X + x_1^2\lambda^2 - 2x_1y_1\lambda + y_1^2 - b$$

We know that $x_1, x_2$ are solutions to Eq

```
# define a polynomial ring for the variables
QQx.<a,b,l,x_1,y_1,x_2,y_2> = QQ[]
QQE.<X,Y> = QQx[]
L = Y - (l*(X-x_1) + y_1); Lx = Y - L
E = Y^2 - X^3 - a*X - b
Eq = E(Y=Lx); Eq # evaluate E at Y = Lx
```

$$-X^3 + \lambda^2 X^2 + (-2x_1\lambda^2 + 2y_1\lambda - a)X + x_1^2\lambda^2 - 2x_1y_1\lambda + y_1^2 - b$$

We know that $x_1, x_2$ are solutions to Eq

`Eq % (X-x_1)` gives $y_1^2 - x_1^3 - ax_1 - b$ this is $E(x_1, y_1)$

```
(Eq % (X-x_1)) %  E(x_1,y_1) == 0
Eq2 = Eq // (X-x_1)
Eq3 = Eq2 % (X-x_2); Eq3
```

$$-\lambda^2 x_1 + \lambda^2 x_2 + 2\lambda y_1 - x_1^2 - x_1x_2 - x_2^2 - a$$

```
# define a polynomial ring for the variables
QQx.<a,b,l,x_1,y_1,x_2,y_2> = QQ[]
QQE.<X,Y> = QQx[]
L = Y - (l*(X-x_1) + y_1); Lx = Y - L
E = Y^2 - X^3 - a*X - b
Eq = E(Y=Lx); Eq # evaluate E at Y = Lx
```

$$-X^3 + \lambda^2 X^2 + (-2x_1\lambda^2 + 2y_1\lambda - a)X + x_1^2\lambda^2 - 2x_1y_1\lambda + y_1^2 - b$$

We know that $x_1, x_2$ are solutions to Eq

Eq % (X-x_1) gives $y_1^2 - x_1^3 - ax_1 - b$ this is $E(x_1, y_1)$

```
(Eq % (X-x_1)) %  E(x_1,y_1) == 0
Eq2 = Eq // (X-x_1)
Eq3 = Eq2 % (X-x_2); Eq3
```

$$-\lambda^2 x_1 + \lambda^2 x_2 + 2\lambda y_1 - x_1^2 - x_1x_2 - x_2^2 - a$$

Substitute $\lambda = (y_2 - y_1)/(x_2 - x_1)$

```
N=QQx(Eq3)(l=(y_2-y_1)/(x_2-x_1)).numerator(); N
```

$$x_1^3 + ax_1 - y_1^2 - x_2^3 - ax_2 + y_2^2$$

```
(N % (E(x_1,y_1))) % E(x_2,y_2) == 0
```

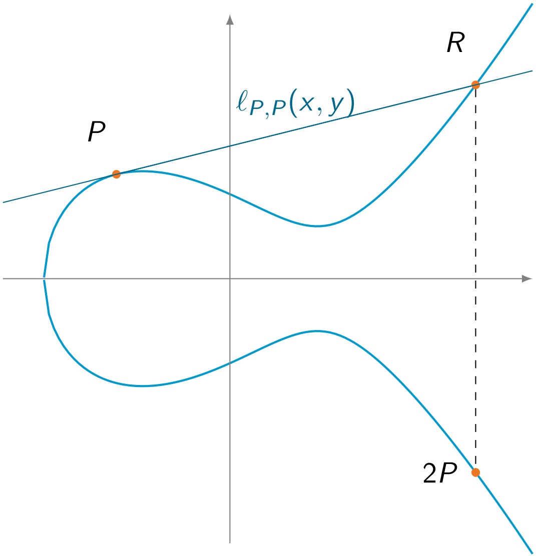
# Chord and tangent rule

```
Eq4 = Eq2 // (X-x_2); Eq4
```

This is $-X + \lambda^2 - x_1 - x_2 = 0 \Rightarrow x_3 = \lambda^2 - x_1 - x_2$
Now $\tilde{y}_3$: $L(x_3, y_3)$: $\tilde{y}_3 = \lambda(x_3 - x_1) + y_1$
Reflect the point $y_3 = -\tilde{y}_3 = -\lambda(x_3 - x_1) - y_1$ Finally,

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

# Doubling a point in affine coordinates $(x, y)$



# Doubling a point in affine coordinates $(x, y)$

The line $L$ tangent at the curve $E\colon f(x,y) = y^2 - x^3 - ax - b = 0$ at $P(x_1, y_1)$ has equation

$$\frac{\partial f}{\partial x}(x_1, y_1) + \frac{\partial f}{\partial y}(x_1, y_1)\frac{dy}{dx} = 0$$

$$\begin{aligned}
(-3x_1^2 - a) + 2y_1\frac{y - y_1}{x - x_1} &= 0 \\
(-3x_1^2 - a)(x - x_1) + 2y_1(y - y_1) &= 0 \\
-\frac{3x_1^2 + a}{2y_1}(x - x_1) + (y - y_1) &= 0 \text{ if } y_1 \neq 0
\end{aligned}$$

The slope is $\lambda = \dfrac{-\partial f/\partial x}{\partial f/\partial y}(x_1, y_1) = \dfrac{3x_1^2 + a}{2y_1}$

Again $L$ has equation $\lambda(x - x_1) + (y - y_1) = 0$

This time we know that $x_1$ is a double root of $E \cap L$

# Doubling a point

```
Eq5 = Eq2 % (X-x_1); Eq5
```

$$-3x_1^2 + 2\lambda y_1 - a$$

```
lambda_dbl = (3*x_1^2 + a)/(2*y_1)
QQx(Eq5)([l=lambda_dbl) == 0
Eq6 = Eq2 // (X-x_1) ; Eq6
```

$$-X + \lambda^2 - 2x_1 \Rightarrow x_4 = \lambda^2 - 2x_1$$

```
x4 = -Eq6.coefficient({X:0,Y:0})/Eq6.coefficient({X:1,Y:0})
Lx([x3,Y])
```

$$y_4 = \lambda(x_4 - x_1) + y_1$$

See `group_law_short_weierstrass_affine.sage`

# Algebraic description of the addition operation

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points on

$$E \colon y^2 = x^3 + ax + b \ .$$

The slope of the line $(P_1, P_2)$ is given by

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq \pm P_2 \\[2ex] \dfrac{3x_1 + a}{2y_1} & \text{if } P_1 = P_2 \text{ and } y_1 \neq 0 \end{cases}$$

The sum of $P$ and $Q$ is the point

$$P + Q = (x_3, y_3) = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1) \ .$$

# Points of order 2, points of order 3

Points of order 2 are such that $P + P = \mathcal{O}$, that is $P = -P$ and $P = (x_0, 0)$.
At $P$ the tangent is a vertical.

Points of order 3 are **inflexion points**.
$2P = -P$ that is the intersection of the tangent at $P$ with the curve is again at $P$, is has multiplicity 3.

# Associativity

Lecture 2.

# Outline

Administrative info

Introduction

Addition Law

Main results

Recap on finite fields

Recap on complexity

# Main questions on curves over $\mathbb{Q}$

Given a bivariate polynomial equation $y^2 = f(x)$ with integer coefficients,

1. Are there any solutions in integers?
2. Are there any solutions in rational numbers?
3. Are there infinitely many solutions in integers?
4. Are there infinitely many solutions in rational numbers?

We will concentrate on these questions for elliptic curves, where

$$y^2 = x^3 + a_2x^2 + a_4x + a_6$$

# Main theorems on curves over $\mathbb{Q}$

A non-singular cubic equation has only finitely many integer solutions (Siegel 1920), bound on the coefficients: Baker–Coates, 1970.

**Nagell–Lutz**: Points of finite order on an elliptic curve have integer coordinates.

**Mordell**: the group of points is finitely generated.

**Mazur**: structure of the group of torsion points (points of finite order)

# Main theorems on curves over $\mathbb{Q}$

## Nagell–Lutz Theorem

Let

$$y^2 = f(x) = x^3 + ax^2 + bx + c$$

be a non-singular cubic curve with integer coefficients $a, b, c$; and let $D$ be the discriminant of the cubic polynomial $f(x)$,

$$= -4a^3c + a^2b^2 + 18abc - 4b^3 - 27c^2 \ .$$

Let $P = (x, y)$ be a rational point of finite order. Then $x$ and $y$ are integers; and either $y = 0$, in which case $P$ has order two, or else $y$ divides $D$.

# Main theorems on curves over $\mathbb{Q}$

## Mazur's theorem

Let $\mathcal{C}$ be a non-singular rational cubic curve, and suppose that $\mathcal{C}(\mathbb{Q})$ contains a point of finite order $m$. Then either

$$1 \leq m \leq 10 \text{ or } m = 12 \ .$$

More precisely, the set of all points of finite order in $\mathcal{C}(\mathbb{Q})$ forms a subgroup which has one of the following two forms:

1. $\mathbb{Z}/n\mathbb{Z}$ A cyclic group of order $n$ with $1 \leq n \leq 10$ or $n = 12$.
2. $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2n\mathbb{Z}$ The product of a cyclic group of order two and a cyclic group of order $2n$ with $1 \leq n \leq 4$.

# Main theorems on curves over $\mathbb{Q}$

### Mordell's theorem (Mordell–Weil)

If a non-singular rational plane cubic curve has a rational point, then the group of rational points is finitely generated.

# Outline

Administrative info

Introduction

Addition Law

Main results

Recap on finite fields

Recap on complexity

# Finite field

**Prime finite field**: a finite field of *prime* order.
(a *prime* field $F$ has no proper non-trivial subfield $K \subsetneq F$)

Notation:

- $\mathbb{Z}/p\mathbb{Z}$: the integers **modulo *p***,
- GF($p$) for Galois Field,
- $\mathbb{F}_p$ (the field of $p$ elements).

Representation: the integers $\{0, 1, 2, \ldots, p-1\}$
or the *centered* set $\{-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2\}$.

The prime number $p$ is the **characteristic** of the finite field.
Field with $p = 2$: $\{0, 1\}$, where $1 + 1 = 0 \bmod 2$
Field with $p = 3$: $\{0, 1, 2\}$ where $1 + 1 = 2$, $1 + 2 = 0 \bmod 3$, $2 + 2 = 1 \bmod 3$
or $\{-1, 0, 1\}$ where $1 + 1 = -1$, $-1 - 1 = 1$, $1 - 1 = -1 + 1 = 0$

# Arithmetic in a prime finite field $\mathbb{F}_p$

### reduction mod $p$

for $x \in \mathbb{Z}$, compute the **Euclidean** division $x = bp + r$ where $0 \leq r < p$. Then $x \bmod p = r$.

### neutral elements

0 is the neutral element for addition, 1 is the neutral element for multiplication

### addition, subtraction $x + y \bmod p$, $x - y \bmod p$

compute $x + y$ as integers, if $x + y \geq p$, subtract $p$
Example: $3 + 5 \bmod 7 = 8 \bmod 7 = 1$

### multiplication: $x \cdot y \bmod p$

Compute $x \cdot y$ like for integers then *reduce* modulo $p$

### inversion

Because $p$ is prime, its **GCD** with any integer $1 \leq x < p$ is 1.
Compute Bézout's identity $ux + vp = 1 = \gcd(x, p)$
Then $ux = 1 \bmod p$ and $1/x = u$

# Extensions of prime fields

What does $\mathbb{F}_{p^2}$ mean? **The** field with $p^2$ elements.
Analogy with the complex numbers $\mathbb{C}$.

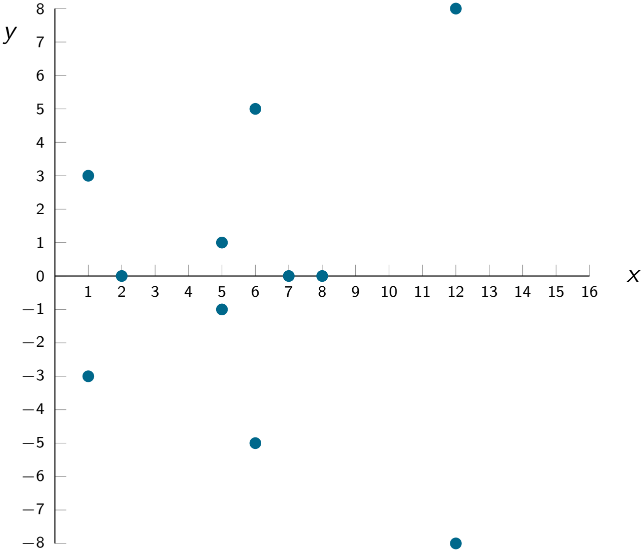
If $p = 3 \bmod 4$, $-1$ is not a square and $X^2 + 1$ is an irreducible polynomial in $\mathbb{F}_p[X]$
Define $\mathbb{F}_{p^2}$ as the quadratic extension $\mathbb{F}_p[X]/(X^2 + 1)$
This notation means: the quotient of all univariate polynomials $a(X)$ with coefficients in $\mathbb{F}_p$, modulo the polynomial $X^2 + 1$.

$X + 5 \bmod (X^2 + 1) = X + 5$
$X^2 \bmod (X^2 + 1) = -1$
$3X^2 + 7X + 1 \bmod (X^2 + 1) = -3 + 7X + 1 = 7X - 2$
$(X + 3) \times (2X - 1) = 2X^2 + 5X - 3 = -2 + 5X - 3 = 5X - 5$

In general, $\mathbb{F}_{p^n}$ is represented as $\mathbb{F}_p[X]/(f(X))$ where $f(X)$ is an irreducible polynomial of degree $n$.

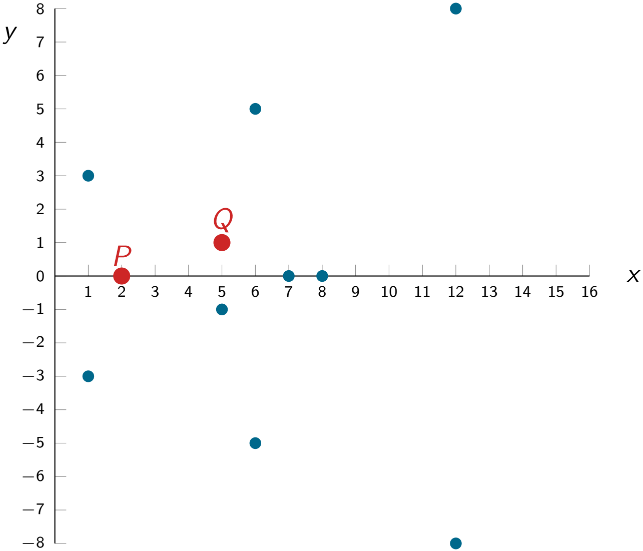
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



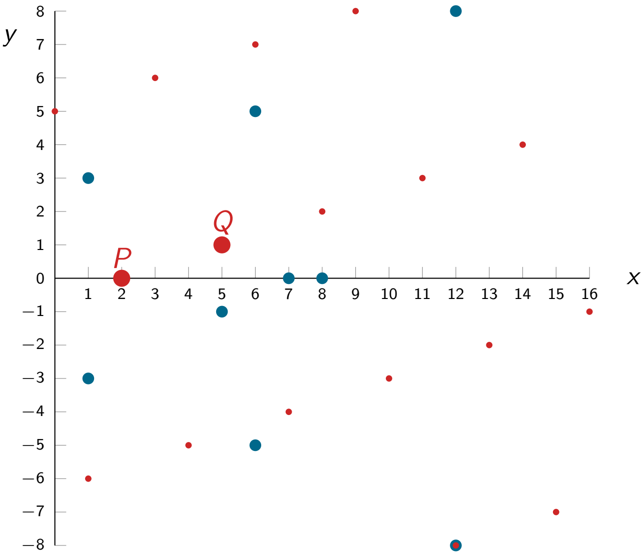
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



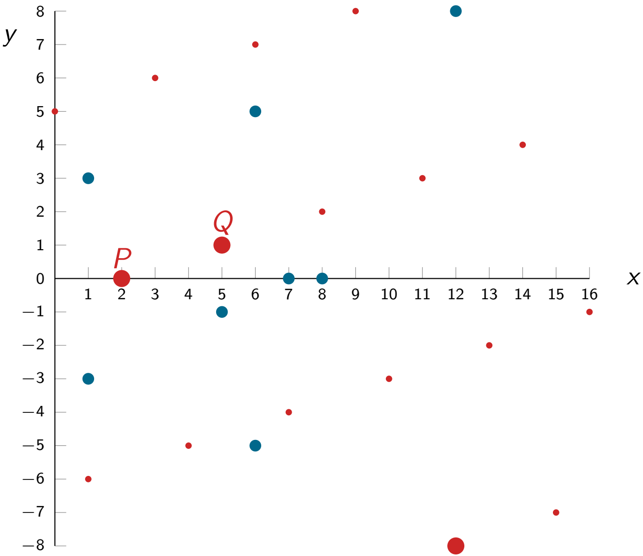
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



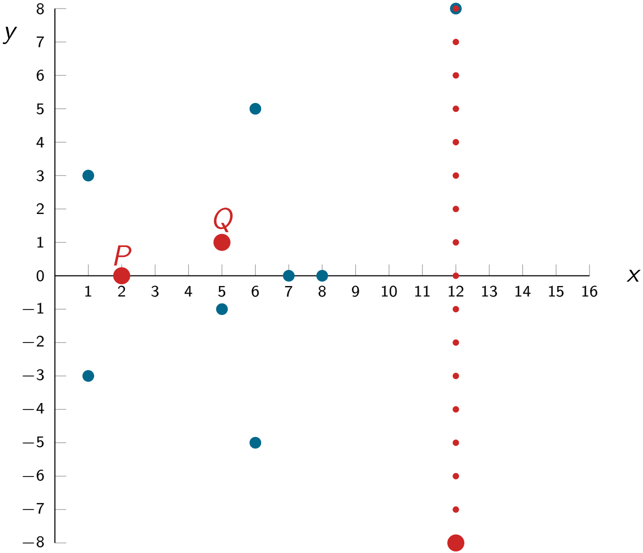
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



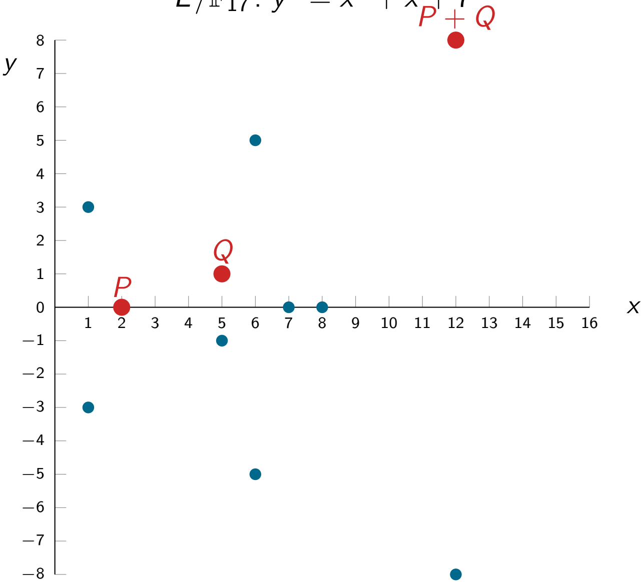
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



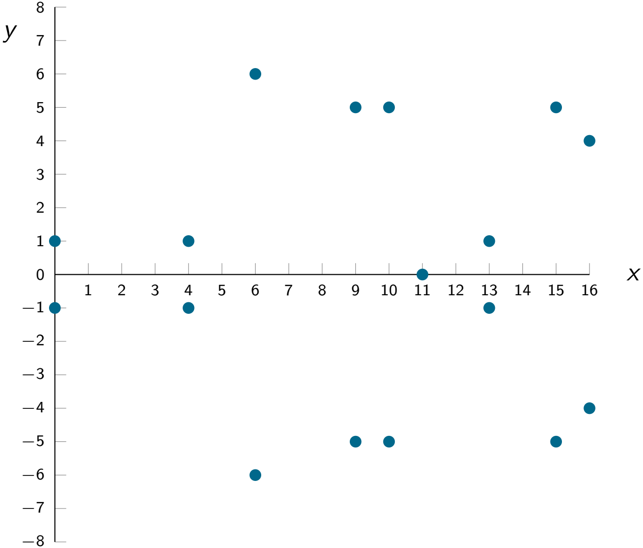
# Elliptic curves over finite fields

$$E/\mathbb{F}_{17}\colon y^2 = x^3 + x + 7$$



# Elliptic curves over finite fields

$$E/\mathbb{F}_{17} \colon y^2 = x^3 + x + 1$$



# Python

How to generate the set of points $(x, y)$ of the curves

- $y^2 = x^3 + x + 7$
- $y^2 = x^3 + x + 1$

over $\mathbb{F}_{17}$? Over $\mathbb{F}_{31}$?

# Outline

Administrative info

Introduction

Addition Law

Main results

Recap on finite fields

Recap on complexity

# Correctness and Complexity of an algorithm

Two important properties of algorithms are correctness and complexity:

- Algorithms should only compute correct solutions of a problem.
  To establish correctness, some relevant logic is introduced.
- What is the time complexity of an algorithm?
- Examples of asymptotic complexity classes of algorithms.

# Efficiency

Given a problem

- Does there exist an algorithm for solving it?
- Does there exist an efficient algorithm?
- Can I improve on a published algorithm for the problem?

Given an algorithm

- How efficient is it?
- Does a more efficient algorithm exist for the same problem?
- Which algorithm for a given problem is the best?

# Two algorithms for summing squares of numbers

```
def sum_square_numbers(n: int):
    """
    Compute the sum of squares up to n
    1^2 +2^2 + 3^2 + ... + n^2
    INPUT:
    - `n`: positive integer
    RETURN: the sum or 0 if n <= 0
    """
    if n <= 0:
        return 0
    s = 1
    i = 2
    while i <= n:
        s = s + i**2
        i = i + 1
    return s
```

```
def sum_square_numbers(n: int):
    """
    Compute the sum of squares up to n
    1^2 +2^2 + 3^2 + ... + n^2
    INPUT:
    - `n`: positive integer
    RETURN: the sum or 0 if n <= 0
    """
    if n <= 0:
        return 0
    return n*(n+1)*(2*n+1)//6
```

Which one is more efficient?

# Measuring efficiency of algorithms

- Cost of running an algorithm depends on size of input (usually)
- Efficiency of an algorithm is expressed as a cost function on the size of input
- Comparison between algorithms difficult to see when size of inputs are small
- Differences in efficiency become apparent as size of input get very large

# Measuring efficiency of algorithms

We need

- a measure on the size of inputs
- a measure of cost of running an algorithm

## Size of inputs:

Dependent on type of data

- arrays: number of items
- lists: number of items
- numbers: often size of binary representation

## Cost of running algorithm:

- time taken – we concentrate on this
- space required – less important these days on computers, matters on embedded devices (smartphones)
- energy consumed – becoming important

# Measuring running time

- Difficult to use a stop watch
- Should not be influenced by speed of computer
- Should not be influenced by choice of programming language
- Should not be influenced by ability of programmer
- We count the number of significant actions

## Significant actions:

Depends on problem area:

- sorting: count comparisons between items
- searching: count comparisons between items
- summing: count arithmetic operations

Sometimes difficult to decide

# Counting the number of operations

```
def sum_square_numbers(n: int):
    """
    Compute the sum of squares up to n
    1^2 +2^2 + 3^2 + ... + n^2
    INPUT:
    - `n`: positive integer
    RETURN: the sum or 0 if n <= 0
    """
    if n <= 0:
        return 0
    s = 1; i = 2
    while i <= n:
        s = s + i**2
        i = i + 1
    return s
```

Operations: $(n-1)$ squares and add.

```
def sum_square_numbers(n: int):
    """
    Compute the sum of squares up to n
    1^2 +2^2 + 3^2 + ... + n^2
    INPUT:
    - `n`: positive integer
    RETURN: the sum or 0 if n <= 0
    """
    if n <= 0:
        return 0
    return n*(n+1)*(2*n+1)//6
```

Operations: 4

# Estimating growth functions

Constants don't count:

| Functions | | $g$ larger than $f$ |
|---|---|---|
| $f$ | $g$ | after |
| $100n^2$ | $2n^3$ | $n > 50$ |
| $1000n^2$ | $3n^3$ | $n > 350$ |
| $1000n^3$ | $n^4$ | $n > 1000$ |

# Estimating growth functions

Small terms get swamped:

| Function | Insignificant after |
|---|---|
| $2n^2 + \underline{10n + 6}$ | $n > 10$ |
| $n^4 + \underline{100n^2 + 5n}$ | $n > 10$ |
| $2n^5 + \underline{1000n^4}$ | $n > 500$ |

# Approximating growth functions

## Big Theta notation

$\Theta(f(n))$: all functions which grow at the same rate as $f(n)$

$g(n)$ is in $\Theta(f(n))$ if

- there is a constant $K_g$ such that $g(n) \leq Kg \cdot f(n)$, once $n$ gets sufficiently large
- there is a constant $K_f$ such that $f(n) \leq K_f \cdot g(n)$, once $n$ gets sufficiently large

which means: once $n$ gets big enough

- $k_1 \cdot f(n) \leq g(n) \leq k_2 \cdot f(n)$
- (and vice-versa)

# Important complexity classes

| | |
|---|---|
| constant | $\Theta(1)$ |
| logarithmic | $\Theta(\log n)$ |
| linear | $\Theta(n)$ |
| $n$-log-$n$ | $\Theta(n \log n)$ |
| quadratic | $\Theta(n^2)$ |
| cubic | $\Theta(n^3)$ |
| polynomial | $\Theta(n^k)$, for some $k \geq 1$ |
| exponential | $\Theta(2^n)$ |

# Examples

- $2 \cdot n + 6$ is in $\Theta(n)$
- $4 \cdot n^2 + 10 \cdot n + 6$ is NOT in $\Theta(n)$
- $234 \cdot n^2 + 658 \cdot n + 200$ is in $\Theta(n^2)$
- $234 \cdot n^2 + 658 \cdot n + 200$ is NOT in $\Theta(n^3)$
- $78 \cdot 10^n + 34 \cdot n^{27}$ is in $\Theta(2^n)$

Dominant exponent always wins out in the end

# Orders of Growth

| $n$ | $\log n$ | $n$ | $n \log n$ | $n^2$ | $n^3$ | $2^n$ |
|---|---|---|---|---|---|---|
| $10$ | $3.3$ | $10$ | $33$ | $100$ | $1,000$ | $1,000$ |
| $100$ | $6.6$ | $100$ | $660$ | $10^4$ | $10^6$ | $1.3 \cdot 10^{30}$ |
| $1,000$ | $10$ | $1,000$ | $10,000$ | $10^6$ | $10^9$ | |
| $10,000$ | $13$ | $10,000$ | $130,000$ | $10^8$ | $10^{12}$ | |
| $100,000$ | $17$ | $100,000$ | $1.7$ million | $10^{10}$ | $10^{15}$ | |
| $1$ million | $20$ | $1,000,000$ | $20$ million | $10^{12}$ | $10^{18}$ | |

Estimated age of the universe: $10^{14}$ seconds

# Particles

| $n$ | $2^n$ | Examples |
|---|---|---|
| 32 | $2^{32} = 10^{9.6}$ | number of humans on Earth |
| 46 | $2^{46} = 10^{13.8}$ | distance Earth - Sun in millimeters<br>number of operations in one day on a processor at 1 GHz |
| 55 | $2^{55} = 10^{16.6}$ | number of operations in one year on a processor at 1 GHz |
| 82 | $2^{82} = 10^{24.7}$ | mass of Earth in kilograms |
| 90 | $2^{90} = 10^{27.1}$ | number of operations in $15 \cdot 10^9$ years (age of the universe) on a processor at 1 GHz |
| 155 | $2^{155} = 10^{46.7}$ | number of molecules of water on Earth |
| 256 | $2^{256} = 10^{77.1}$ | number of electrons in universe |

# Boiling water

Universal Security; From bits and mips to pools, lakes – and beyond
Arjen Lenstra, Thorsten Kleinjung, and Emmanuel Thomé
`https://hal.inria.fr/hal-00925622`

- $2^{90}$ operations require enough energy to boil the lake of Genève
- $2^{114}$ operations: boiling all the water on Earth
- $2^{128}$ operations: boiling 16000 planets like the Earth

# Why buying a new computer will not help

Running an exponential algorithm

Current computer: 10,000 instructions per second

| problem size | time |
|---|---|
| 10 | 0.1 sec |
| 20 | 2 mins approx. |
| 30 | $>$ 24 hours |
| 38 | $>$ one year |

# Why buying a new computer will not help

Running an exponential algorithm

New computer: 100 times faster

| problem size | time |
|---|---|
| 10 | 0.1 sec |
| 20 | 1 minute approx |
| 37 | $>$ 24 hours |
| 45 | $>$ one year |

Only minimal increase in effectiveness:

| time T | problem size |
|---|---|
| old computer | $n$ |
| new computer | $n + 7$ |

Similar phenomenon for polynomial algorithms – But less pronounced

# Algorithms v. fast computers

More efficient algorithms better than faster computers

Setup A:

- slow sorting algorithm: $\Theta(n^2)$
- crafty programmer: constant factor 2
- fast machine: 1 billion instructions per second

Setup B:

- fast sorting algorithm: $\Theta(n \log(n))$
- rubbish programmer: constant factor 50
- slow machine: 10 million instructions per second (100 times slower)

| setup | 1 million numbers | 10 million numbers |
|---|---|---|
| A | $> 30$ mins | $> 2$ days |
| B | $< 2$ mins | $< 20$ mins |

# Comparing complexity classes

| Complexity class | Extra time required when doubling the size of input |
|---|---|
| $\Theta(1)$ | none |
| $\Theta(\log n)$ | marginal increase |
| $\Theta(n)$ | double |
| $\Theta(n \log n)$ | double + tiny |
| $\Theta(n^2)$ | four times longer |
| $\Theta(n^3)$ | eight times longer |
| $\Theta(2^n)$ | square of time |

# Characteristics

| Class | Name | Characteristics |
|---|---|---|
| $\Theta(1)$ | constant | few interesting algorithms |
| $\Theta(\log n)$ | logarithmic | result of cutting problem size in half each time round a loop |
| $\Theta(n)$ | linear | Algorithms which scan an array/list |
| $\Theta(n \log n)$ | $n$-log-$n$ | Many divide-and-conquer algorithms |
| $\Theta(n^2)$ | quadratic | Algorithms with two embedded loops |
| $\Theta(n^3)$ | cubic | Algorithms with three embedded loops |
| $\Theta(2^n)$ | exponential | Many important problems |

# Next Lectures

## Office hours, Tuesday, February 1, room 5335-395 at Nygaard building (CS)

- Help on Installing SageMath

## Tutorial 1, Thursday, February 3, room 1532-314 (Maths)

- Installing SageMath, starting to run small programs
- Bachet replication formula with SageMath

## Lecture 2, Tuesday, February 8

- Projective space $\mathbb{P}^3$ and the point at infinity
- Addition law in projective coordinates
- The law is associative
- Intersection multiplicity and Bézout's theorem

# Credits

- Jérémie Detrey for many slides and support from ARCHI'2017 summer school
- Laurent Imbert for slides from ECC'11 summer school
- Ian Mackie and Marine Minier for the recap on complexity of algorithms
- Simon Masson for the graph on page 22 from his PhD thesis
- Christophe Ritzenthaler for ressources at his webpage
- Emmanuel Thomé en Cyril Bouvier for slides from a winter school at ISI Delhi in 2017
- Ben Smith for his slides from MPRI