Algorithms for finite field arithmetic

Éric Schost
(joint with Luca De Feo & Javad Doliskani)

Western University → University of Waterloo

July 9, 2015

Basics

Finite fields

Definition

A finite field is a **field** (a set with addition, multiplication, inverse) which is **finite**.

Examples.

- $\mathbb{F}_2 = \{0, 1\}$, with operations XOR and AND
- $\mathbb{F}_p = \{0, \ldots, p-1\}$, p prime, with addition and multiplication mod p
- $\{0, 1, 2, 3\}$ with operations mod 4 is not a field

Finite fields are ubiquitous

- number theory and algebraic geometry
- cryptography
  - elliptic curve cryptography
  - multivariate cryptography
- coding theory
  - Reed-Solomon, AG codes, ...

Our objective

Efficient algorithms for building and working with finite fields

Computing in finite fields

```
k1:=GF(5^1);
a1:=Random(k1);
...

k10:=GF(5^10);
a10:=Random(k10);
...

k100:=GF(5^100);
a100:=Random(k100);
...

k1000:=GF(5^1000);
a1000:=Random(k1000);
...
```

How does this scale?

Building finite fields

If Q is an irreducible polynomial of degree d over $\mathbb{F}_p$,

$$\mathbb{F}_p[X]/Q(X) = \{a_0 + a_1 X + \cdots + a_{d-1} X^{d-1} \mid a_i \in \mathbb{F}_p\}$$

is a finite field with p^d elements, with operations done mod p and Q.

Facts:

- all finite fields can be constructed this way
- no canonical choice

Not covered here

finding primes, normal bases, Zech logarithms, Conway polynomials, ...

On the algorithmic side

Basic arithmetic

- operands: elements of $\mathbb{F}_p$
- operations $+, \times, \div$ in $\mathbb{F}_p$ have unit cost

Working with $\mathbb{F}_{p^d}$

- an element of $\mathbb{F}_{p^d}$: d elements of $\mathbb{F}_p$
- polynomial time: $(d \log(p))^{O(1)}$

If $Q(X)$ is given

Arithmetic in $\mathbb{F}_{p^d}$ is **easy**:

- operations on univariate polynomials (multiplication, division, XGCD)
- quasi-linear time if FFT-based techniques are used.

The big picture

No deterministic polynomial-time algorithm is known.

- Deterministic algorithms run in time $(dp)^{O(1)}$ [Shoup'89]
- With $d = 2$, this amounts to finding x in $\mathbb{F}_p$ which is not a square
  - $O(1)$ random choices suffice
  - under Generalized Riemann Hypothesis, $\log(p)^{O(1)}$ choices
- Same ideas in higher degrees
  - ERH [Adleman-Lenstra]
  - recent work by [Ivanyos et al.] to remove dependency on GRH.

Not covered here

Special primes [von zur Gathen, Rónyai, Shoup], average case analysis [Gao-Panario], bounds on degrees [von zur Gathen, Adleman-Lenstra], ...

Lattices of finite fields

Computing in finite fields

A Magma session:

```
k4:=GF(5^4);
k6:=GF(5^6);
a4:=Random(k4);
a6:=Random(k6);
a:=a4+a6;
Parent(a);
 Finite field of size 5^12
```

The question is not only building $\mathbb{F}_{5^4}$ or $\mathbb{F}_{5^6}$. We also have to make them all fit together.

More on finite fields

Fact: if m divides n, there is an embedding $\mathbb{F}_{p^m} \hookrightarrow \mathbb{F}_{p^n}$.

For instance,

$$\mathbb{F}_p \hookrightarrow \mathbb{F}_{p^2} \hookrightarrow \mathbb{F}_{p^4} \hookrightarrow \mathbb{F}_{p^8} \cdots$$

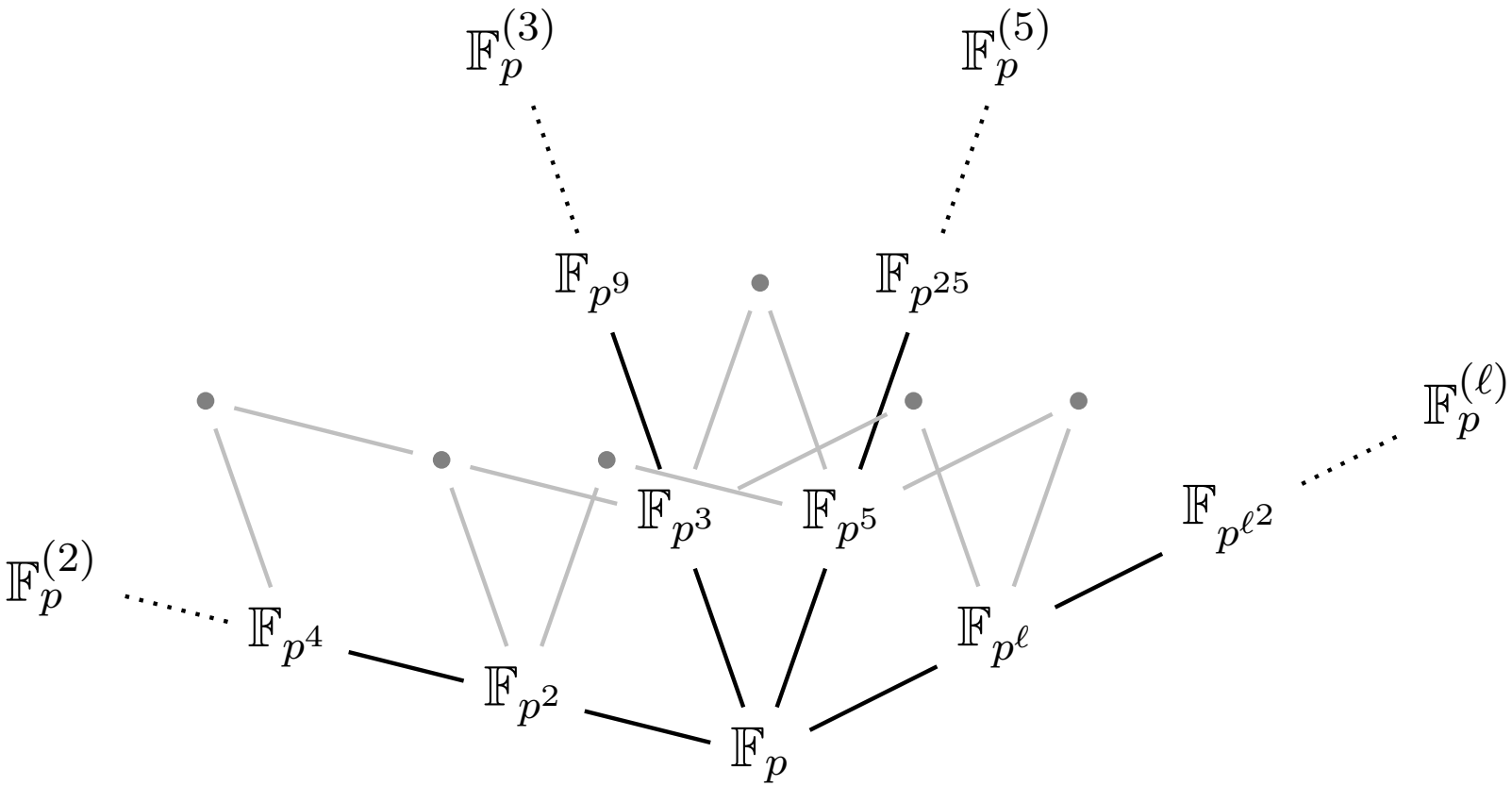
is obtained by a series of extensions of degree 2. Same with powers of $3, 5, \ldots$

Explicitly

Amounts to the following computation:

$$\begin{array}{ccc} \mathbb{F}_p[X]/Q(X) & \hookrightarrow & \mathbb{F}_p[X]/R(X) \\ F(X) & \mapsto & F(G) \bmod R. \end{array}$$

What does $\bar{\mathbb{F}}_p$ look like?



From [De Smit-Lenstra]

Some previous work

All in a similar spirit:

- [Shoup'90] and [Shoup'94] — irreducibles
- [Couveignes-Lercier] — irreducibles
- [De Smit-Lenstra] — standard model

Very complete design in Magma [Bosma-Cannon-Steel]

- several representations and algorithms
- arbitrary field isomorphisms and embeddings

Libraries:

- PARI, NTL, FLINT, ...

Interlude:
Polynomial arithmetic

Univariate and multivariate

An extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[X]/\langle X^6 + 4X^5 + 2X^4 + 5X^2 + 9X + 6\rangle$$

Univariate and multivariate

An extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[X]/\langle X^6 + 4X^5 + 2X^4 + 5X^2 + 9X + 6\rangle$$

Another extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[Z,T]/\langle Z^3 + 3Z^2 + 5Z + 1, T^2 + 6T + 1\rangle$$

Univariate and multivariate

An extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[X]/\langle X^6 + 4X^5 + 2X^4 + 5X^2 + 9X + 6\rangle$$

Another extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[Z,T]/\langle Z^3 + 3Z^2 + 5Z + 1, T^2 + 6T + 1\rangle$$

Working in the second model

- multiplication: reduction by two polynomials
- inversion: similar to XGCD, more complex

Univariate and multivariate

An extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[X]/\langle X^6 + 4X^5 + 2X^4 + 5X^2 + 9X + 6\rangle$$

Another extension of degree 6 of $\mathbb{F}_{11}$:

$$\mathbb{F}_{11}[Z,T]/\langle Z^3 + 3Z^2 + 5Z + 1, T^2 + 6T + 1\rangle$$

A useful tool: change-of-basis.

$$F(X) \mapsto F(T^2Z + 3T^2 + 10TZ + 8T + 2Z + 9) \bmod \langle Z^3 + \cdots, T^2 + \cdots\rangle$$

Triangular sets

Continuing this way, we may end up with structures such as

$$\left|\begin{array}{l} T_n(X_1, \ldots, X_n) \\ \quad \vdots \\ T_2(X_1, X_2) \\ T_1(X_1) \end{array}\right.$$

Triangular sets

- many algorithms for polynomial system solving [Ritt, Wu, Lazard, Kalkbrenner, Moreno Maza, . . .]
- still no quasi-linear algorithm for basic arithmetic

Triangular sets

Continuing this way, we may end up with structures such as

$$\left|\begin{array}{l} T_n(X_1, \ldots, X_n) \\ \quad \vdots \\ T_2(X_1, X_2) \\ T_1(X_1) \end{array}\right.$$

Triangular sets

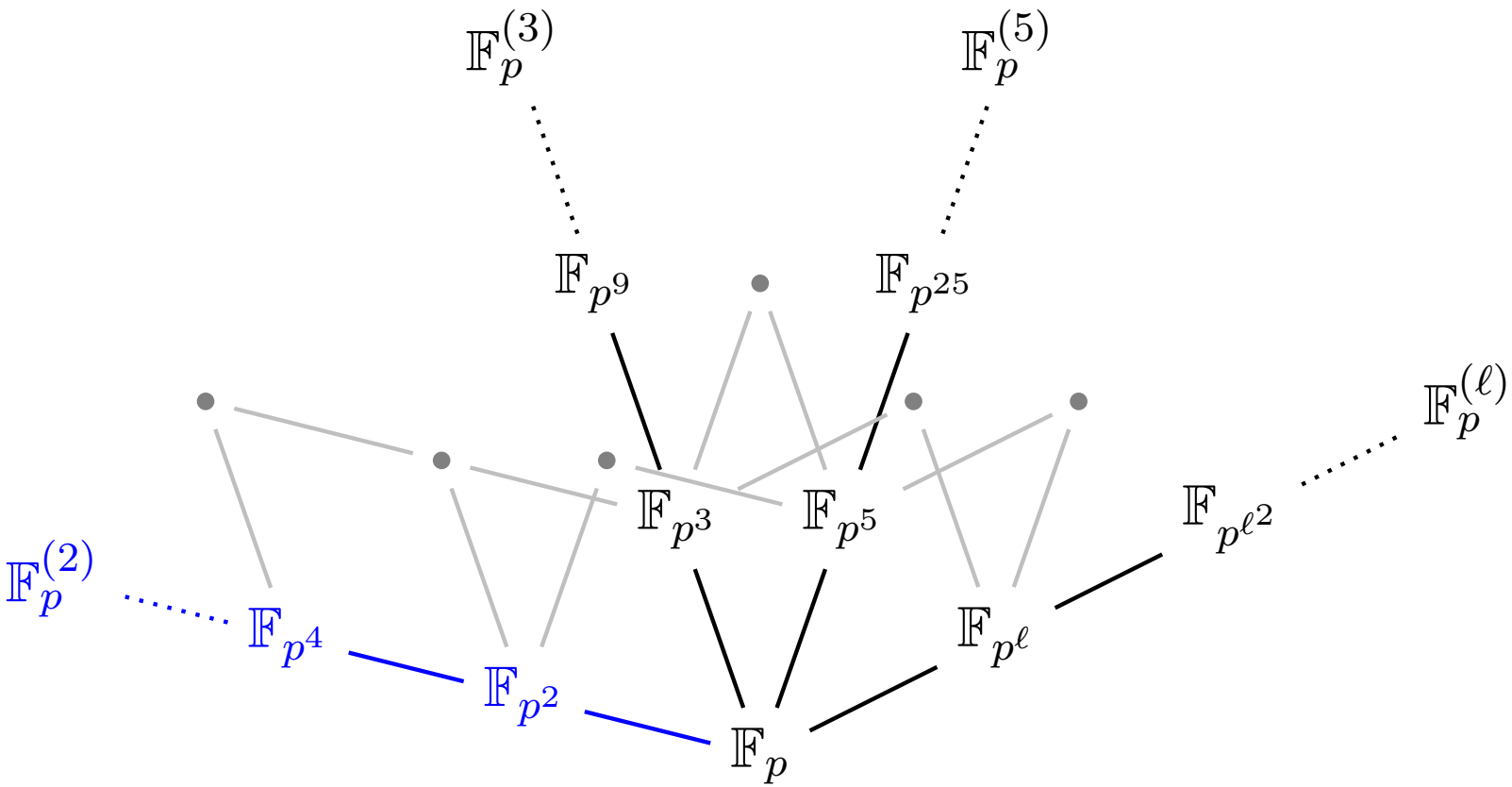
- many algorithms for polynomial system solving [Ritt, Wu, Lazard, Kalkbrenner, Moreno Maza, . . .]
- still no quasi-linear algorithm for basic arithmetic

Change of basis

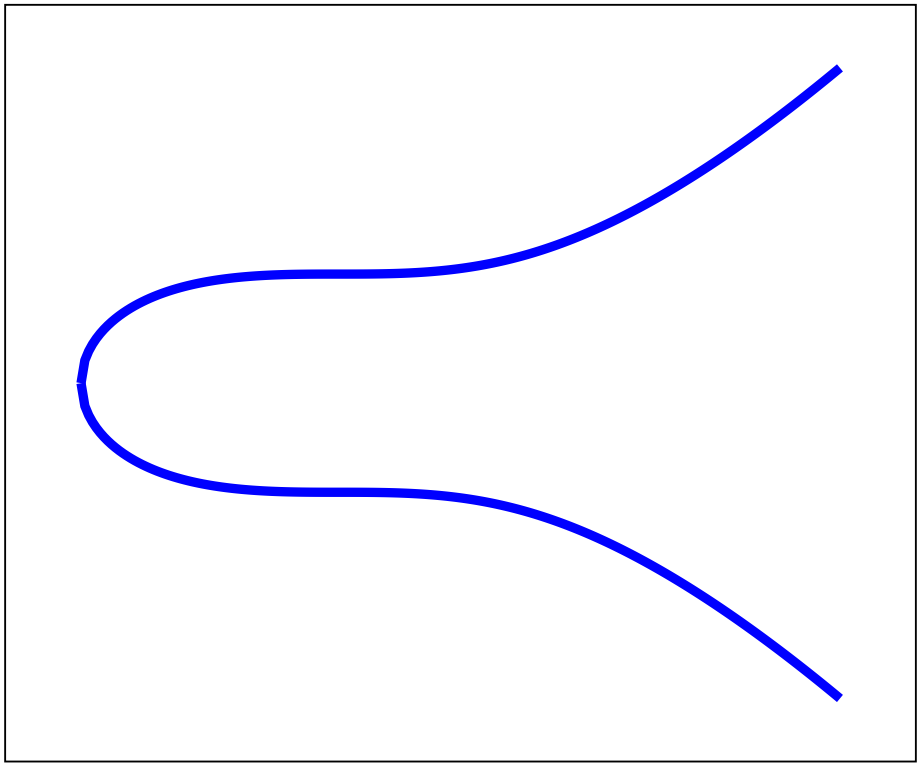
- almost linear time [Umans, Kedlaya-Umans, Poteaux-S.] in a boolean model
- does not appear to be useful in practice (yet)

Towers

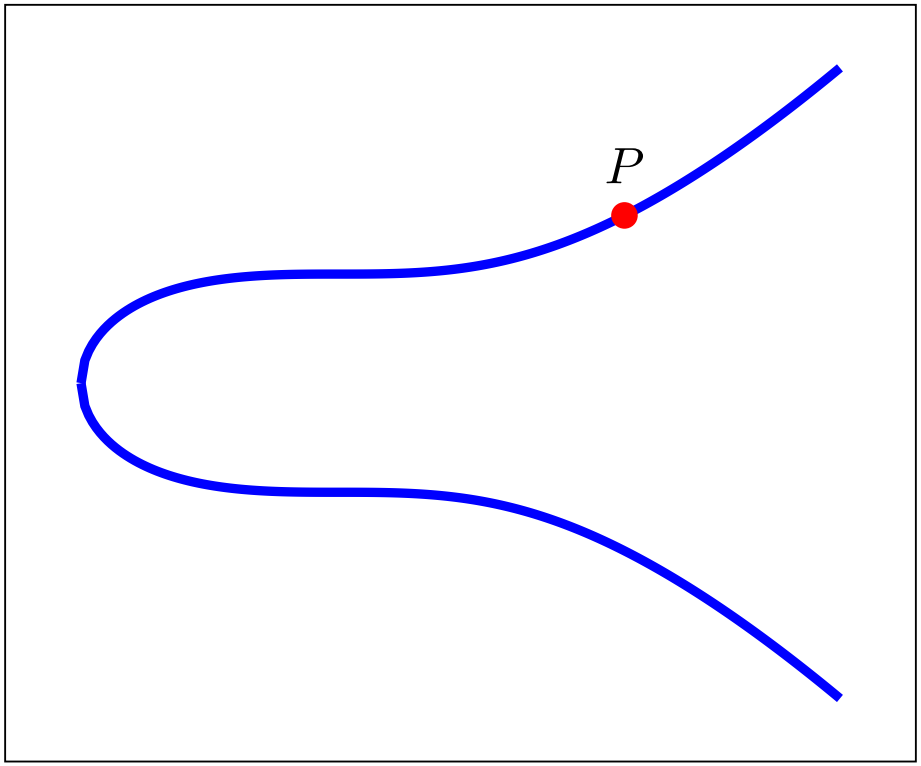
One direction of the lattice



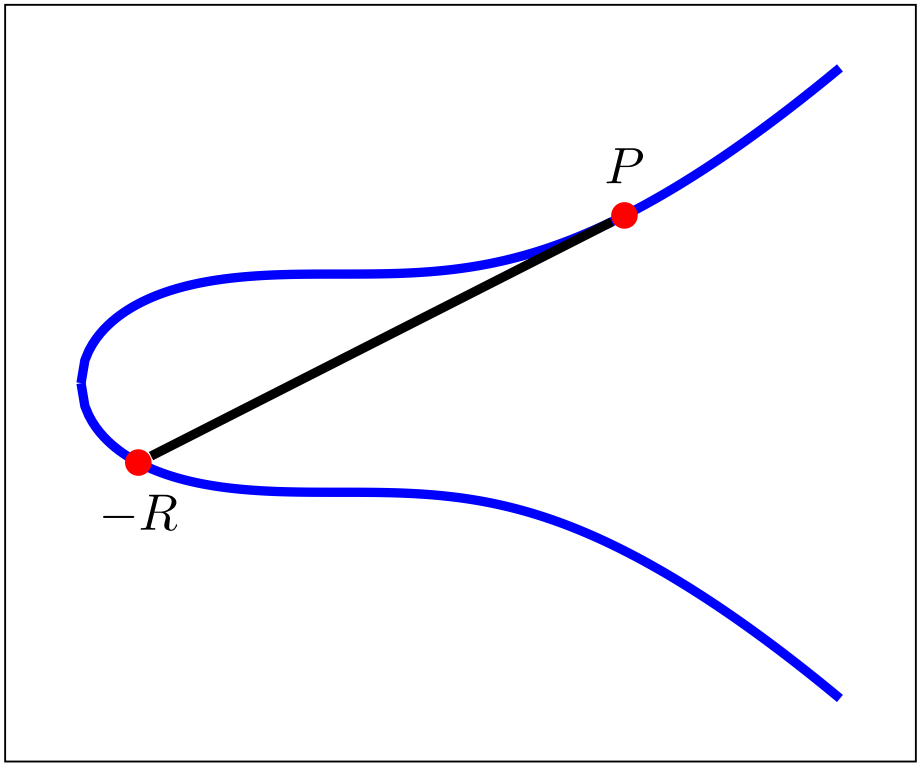
Example: halving on an elliptic curve



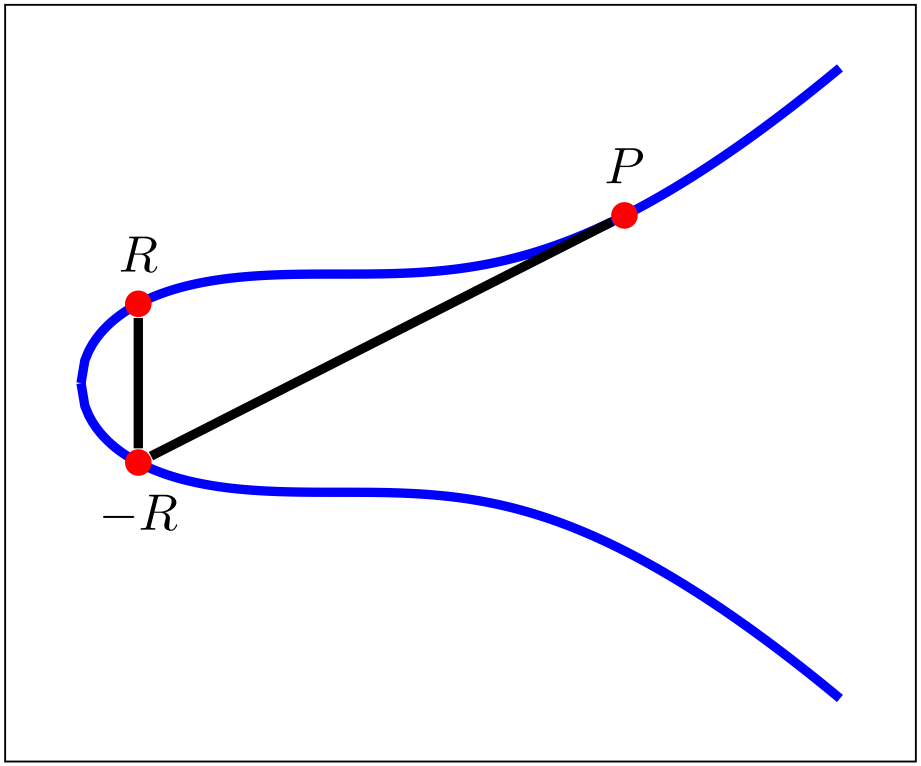
Example: halving on an elliptic curve



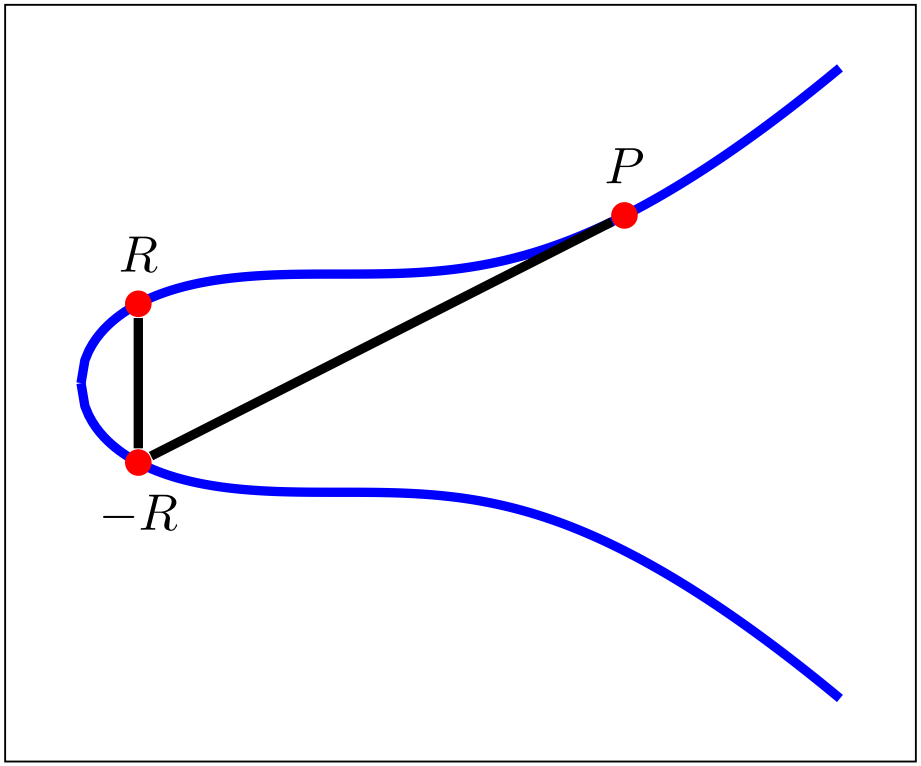
Example: halving on an elliptic curve



Example: halving on an elliptic curve

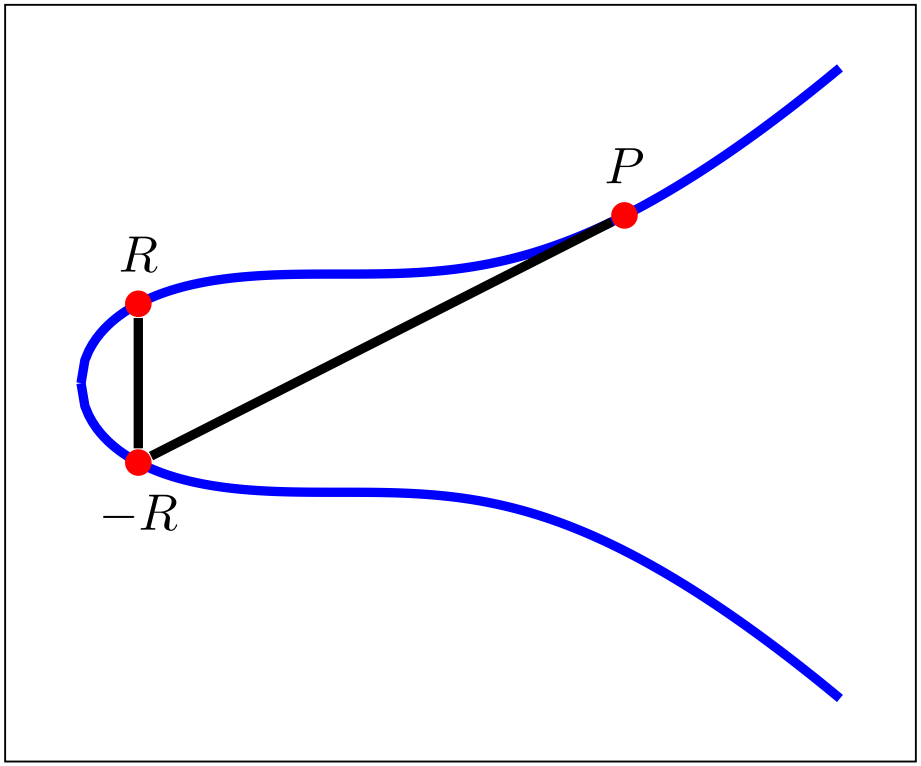


Example: halving on an elliptic curve



Recovering P from $R \rightarrow$ extracting 2 square roots

Example: halving on an elliptic curve



Recovering P from $R \rightarrow$ extracting 2 square roots

Similar questions

- division by p [Couveignes, De Feo]
- hyperelliptic curves [Gaudry-S.]

Basic construction

Smallest prime, $\ell = 2$:

- Suppose that x_0 is not a square. Then $X^2 - x_0$ is irreducible. And $X^4 - x_0$. And $X^8 - x_0 \ldots$ $\qquad p \equiv 1 \pmod 4$

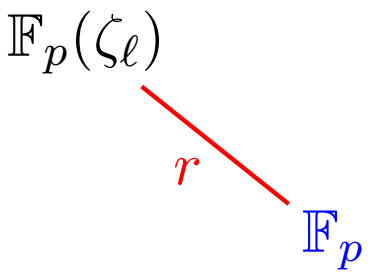
General prime ℓ, with $\ell \neq p$:

- If x_0 is not an ℓth power, $X^{\ell^i} - x_0$ is irreducible for all i.
- Existence of $x_0 \iff$ existence of ℓth roots of unity $\iff$ ℓ divides $p - 1$.

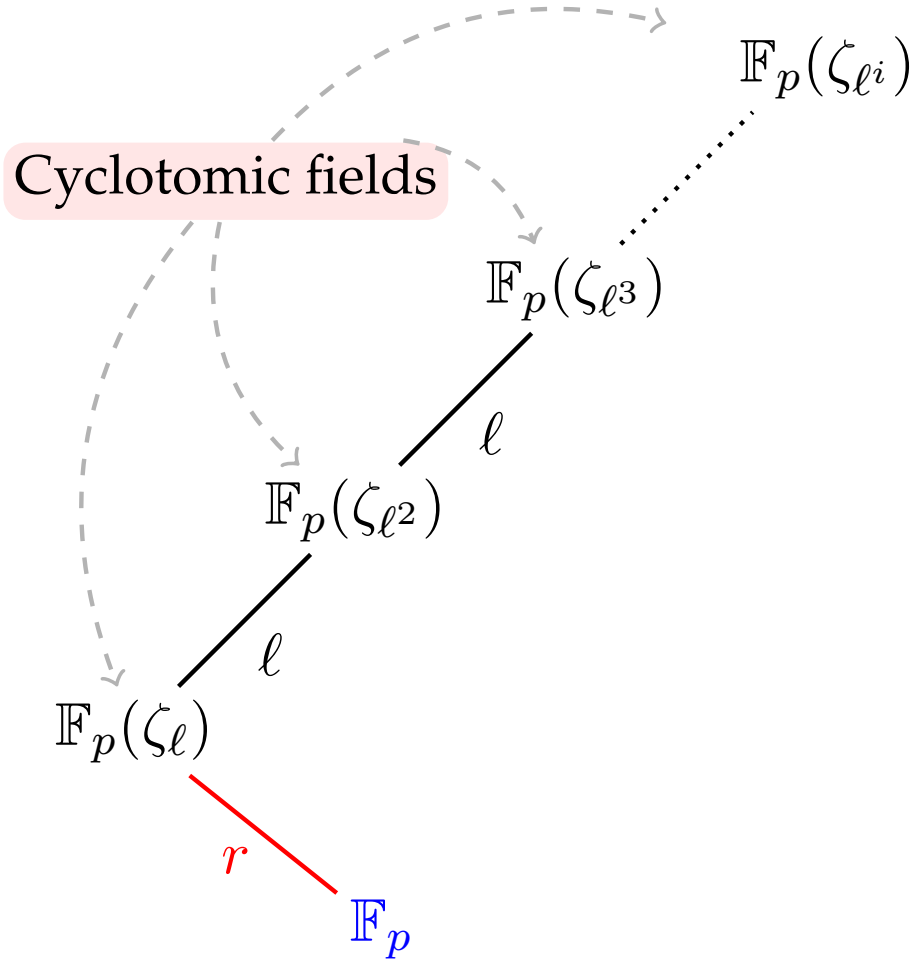
We are looking at fibers of $x \mapsto x^\ell$

Cyclotomy [Shoup, De Smit-Lenstra, De Feo-Doliskani-S.]

replace $\mathbb{F}_p$ by $\mathbb{F}_p(\zeta_\ell)$
(ζ_ℓ: ℓth root of unity)

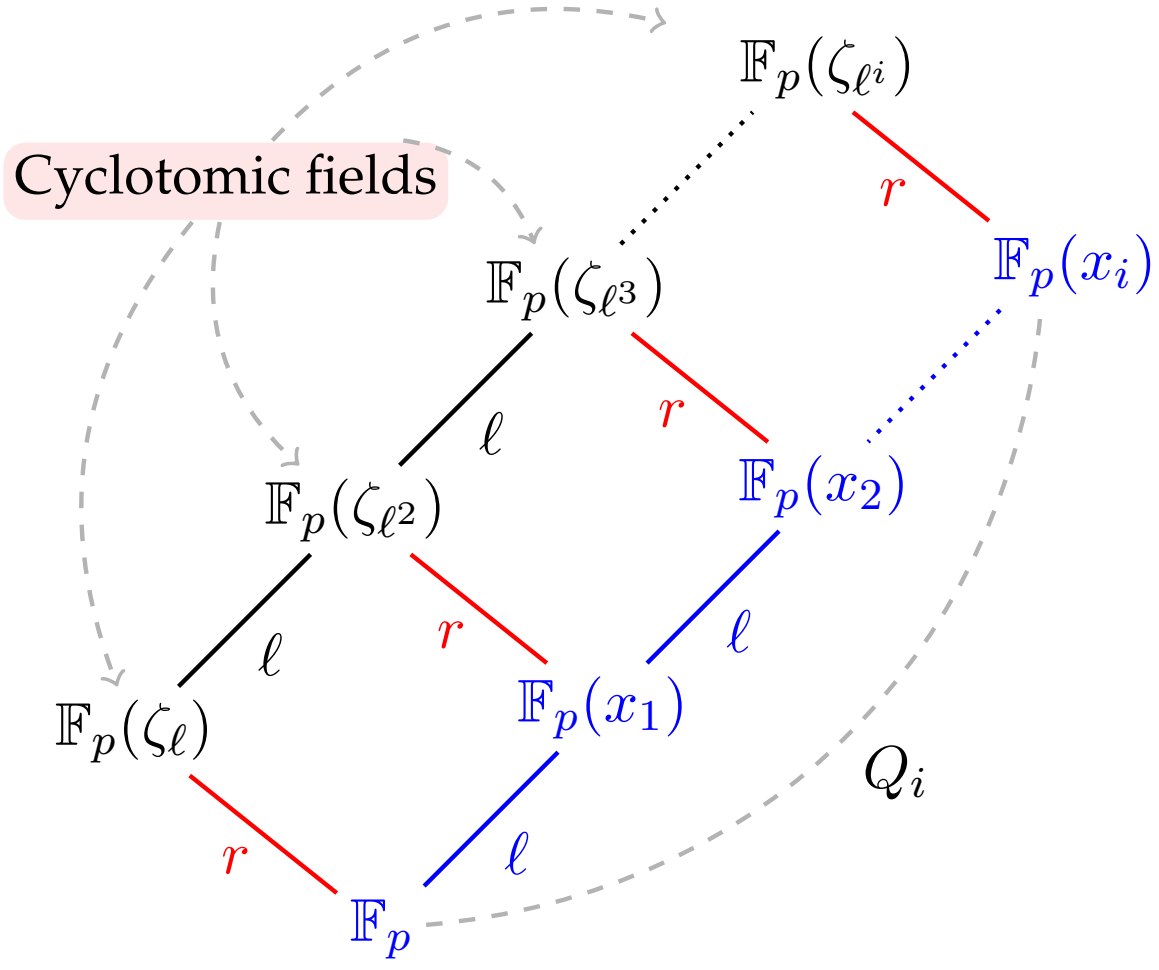


Cyclotomy [Shoup, De Smit-Lenstra, De Feo-Doliskani-S.]



do as before

Cyclotomy [Shoup, De Smit-Lenstra, De Feo-Doliskani-S.]



- Q_i can be computed by resultants
- divide-and-conquer algorithm for embedding
- cost: $\tilde{O}(\ell^{i+c})$

Elliptic curves [Couveignes-Lercier, De Feo-Doliskani-S.]

Rule of thumb

If you know an algorithm relying on cyclotomic constructions, it may have an elliptic counterpart:

$$\text{multiplication in } \mathbb{F}_p^\star \quad \longleftrightarrow \quad \text{addition on an elliptic curve}$$

Examples:

- Pollard's $p-1$ and extensions / Lenstra's ECM
- Primality test, FFT, ...

Elliptic curves [Couveignes-Lercier, De Feo-Doliskani-S.]

Rule of thumb

If you know an algorithm relying on cyclotomic constructions, it may have an elliptic counterpart:

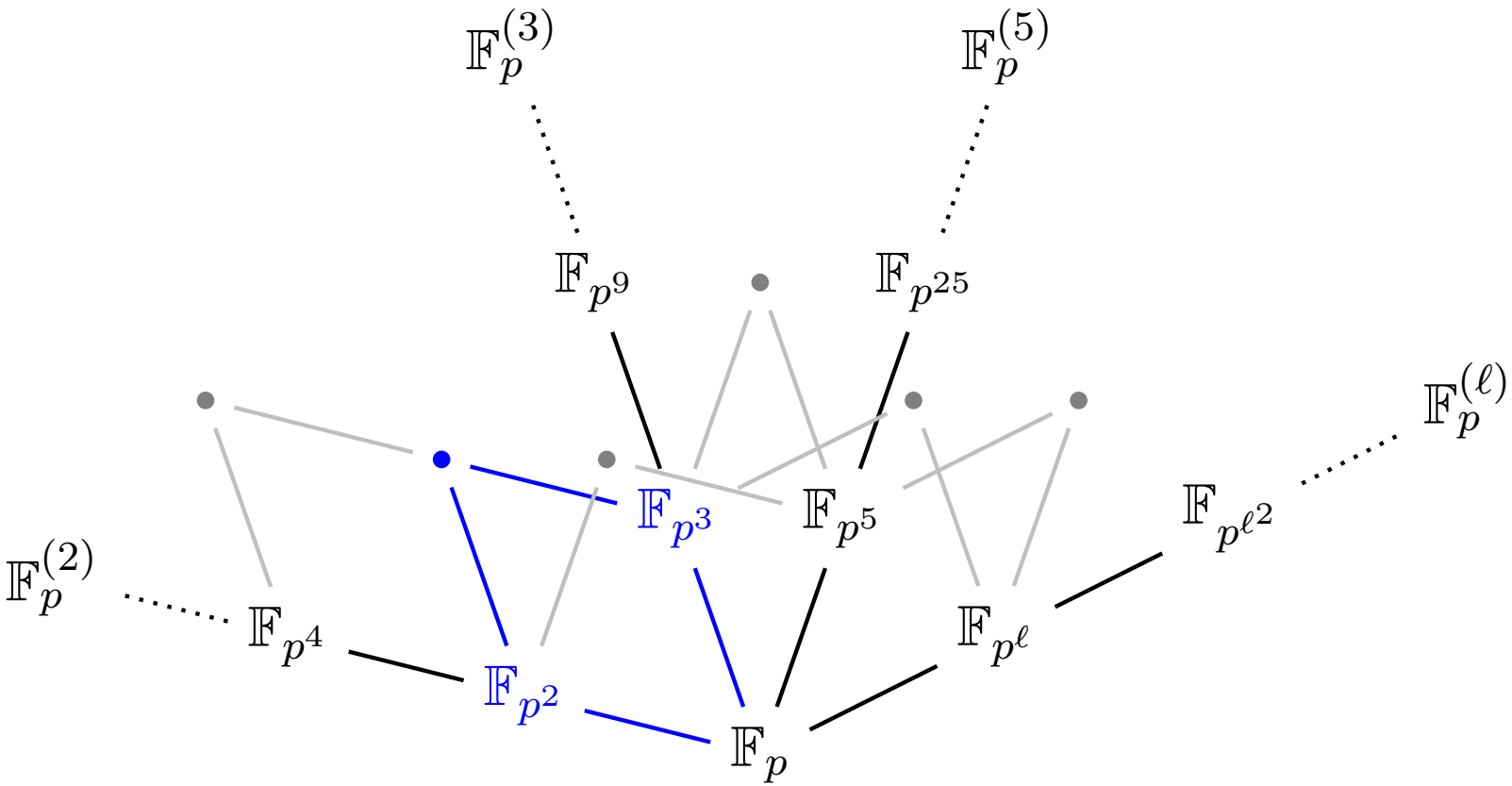
$$\text{multiplication in } \mathbb{F}_p^\star \quad \longleftrightarrow \quad \text{addition on an elliptic curve}$$

Here:

- we use an analogue of the ℓth-power map $x \mapsto x^\ell$
  (isogenies between curves)
- need curves with suitable cardinality properties
- divide-and-conquer algorithm for embedding, cost $\tilde{O}(\ell^{i+c})$

Completing the lattice

What is left to do



Composita of fields

Composed product [Brawley-Carlitz]

Suppose that

$$P = \prod_{i=1,\ldots,m} (X - a_i) \quad \text{and} \quad Q = \prod_{j=1,\ldots,n} (X - b_j)$$

Their composed product is

$$R = \prod_{i,j} (X - a_i b_j).$$

Prop.

- if m and n are coprime, over a finite field, R is irreducible

Composita of fields

Composed product [Brawley-Carlitz]

Suppose that

$$P = \prod_{i=1,\ldots,m} (X - a_i) \quad \text{and} \quad Q = \prod_{j=1,\ldots,n} (X - b_j)$$

Their composed product is

$$R = \prod_{i,j} (X - a_i b_j).$$

Computing R

- as a resultant [Shoup]
- quasi-linear through its Newton sums [Bostan et al.]

Change of basis

Given

$$x = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad y = \sqrt[3]{3},$$

how to find that $x = \frac{1}{6}(xy)^3 + \frac{1}{2}$? This is linear algebra.

Change of basis

Given

$$x = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad y = \sqrt[3]{3},$$

how to find that $x = \frac{1}{6}(xy)^3 + \frac{1}{2}$? This is linear algebra.

Key idea

Turning a question about matrices to a question about sequences:

- sparse linear algebra [Wiedemann]
- sparse FGLM [Faugère *et al.*]
- RUR [Rouillier]

Change of basis

Given

$$x = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad y = \sqrt[3]{3},$$

how to find that $x = \frac{1}{6}(xy)^3 + \frac{1}{2}$? This is linear algebra.

Key idea

Turning a question about matrices to a question about sequences:

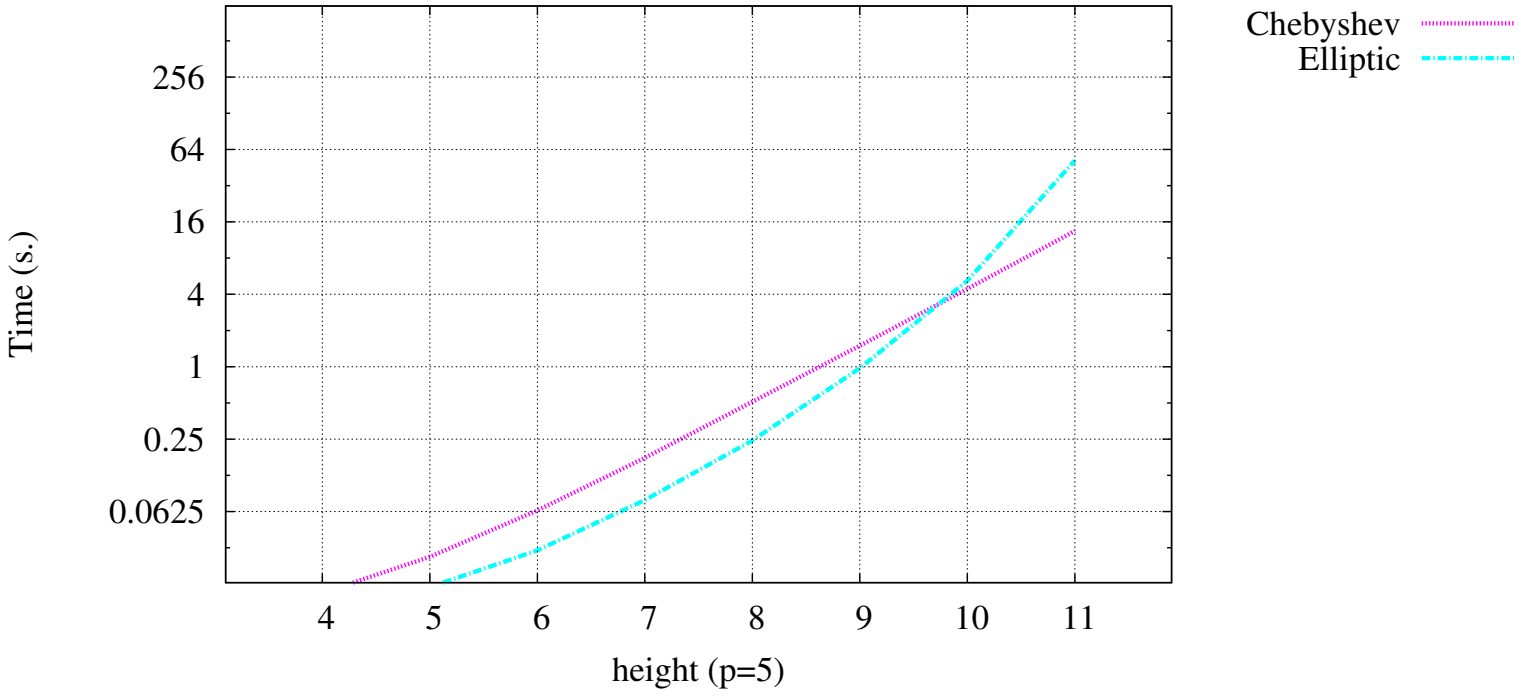
- sparse linear algebra [Wiedemann]
- sparse FGLM [Faugère *et al.*]
- RUR [Rouillier]

Our results [De Feo-Doliskani-S.]

- embeddings $\mathbb{F}_{p^m} \hookrightarrow \mathbb{F}_{p^{mn}}$ quasi-linear

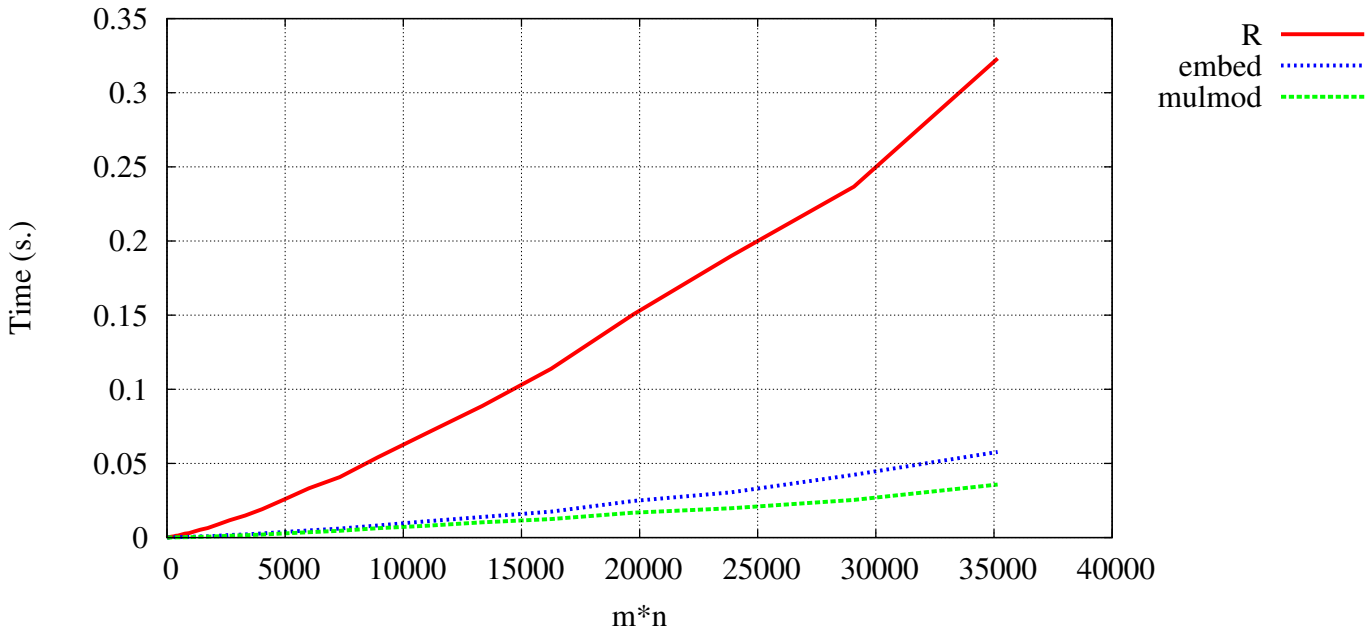
Conclusions

Sage / FLINT implementation



- Times for building 3-adic towers on top of $\mathbb{F}_5$
- Intel Xeon E5620 clocked at 2.4 GHz, using Sage 5.5
- Source code at `https://github.com/defeo/towers`.

Sage / FLINT implementation



- times for embedding in degree mn, with $m = n + 1$.
- Source code at `https://github.com/defeo/ff_compositum`.

Conclusion

Results

- many algorithms, several of which are linear time
- some still need to be implemented

Loose ends and further work

- make everything linear time
- revisit isomorphisms