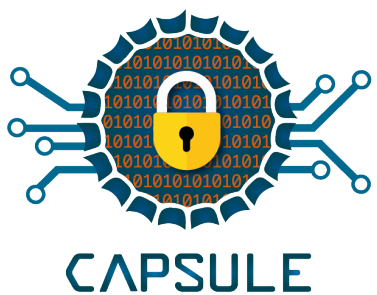
Finding e-th roots in a number field

Olivier Bernard, Pierre-Alain Fouque, Andrea Lesavourey

Univ Rennes, CNRS, IRISA

March 6, 2023



Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

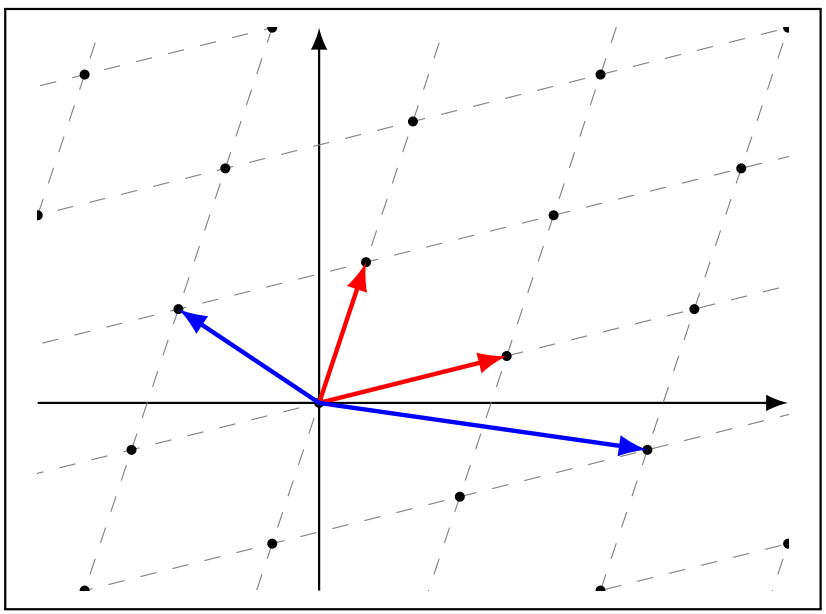
2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Euclidean lattices

General context

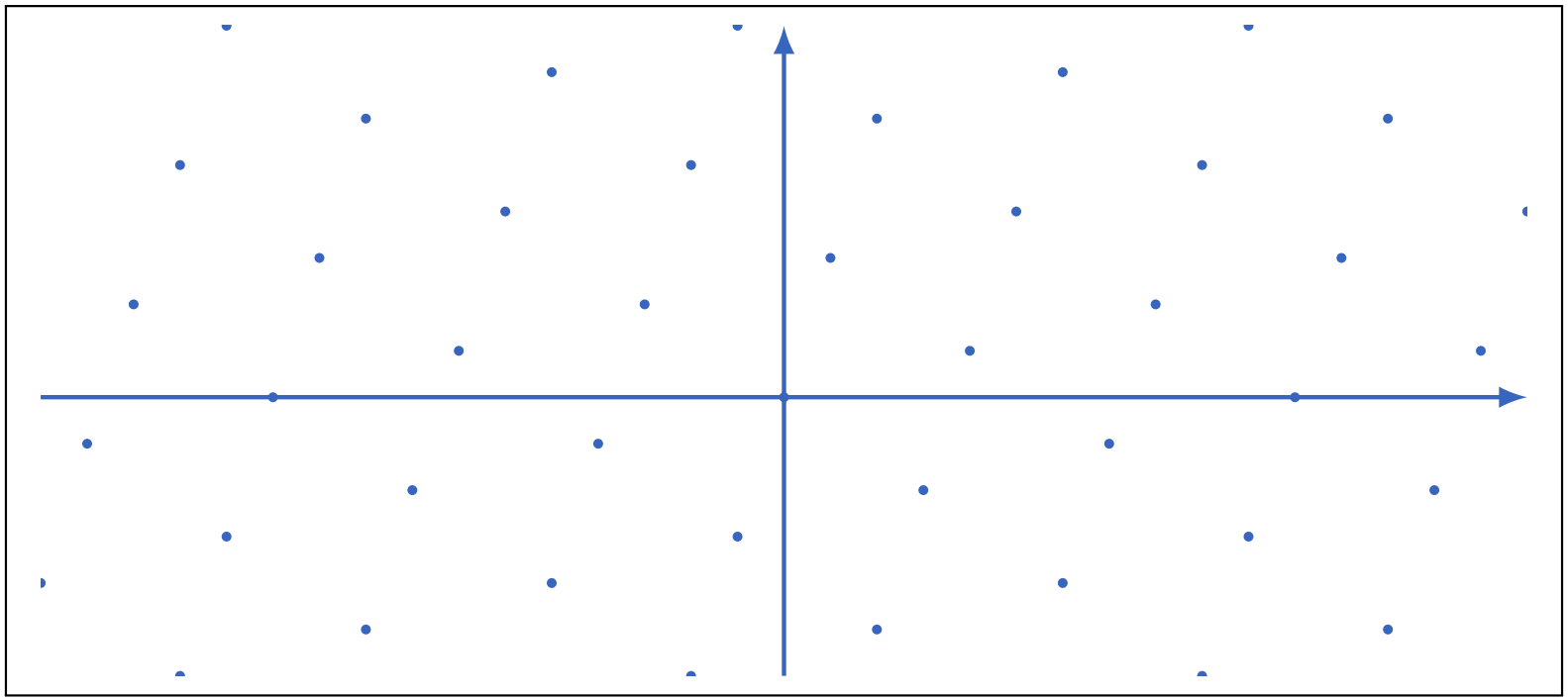
Definition

We call *lattice* any discrete subgroup $\mathcal{L}$ of $\mathbb{R}^n$ where n is a positive integer.

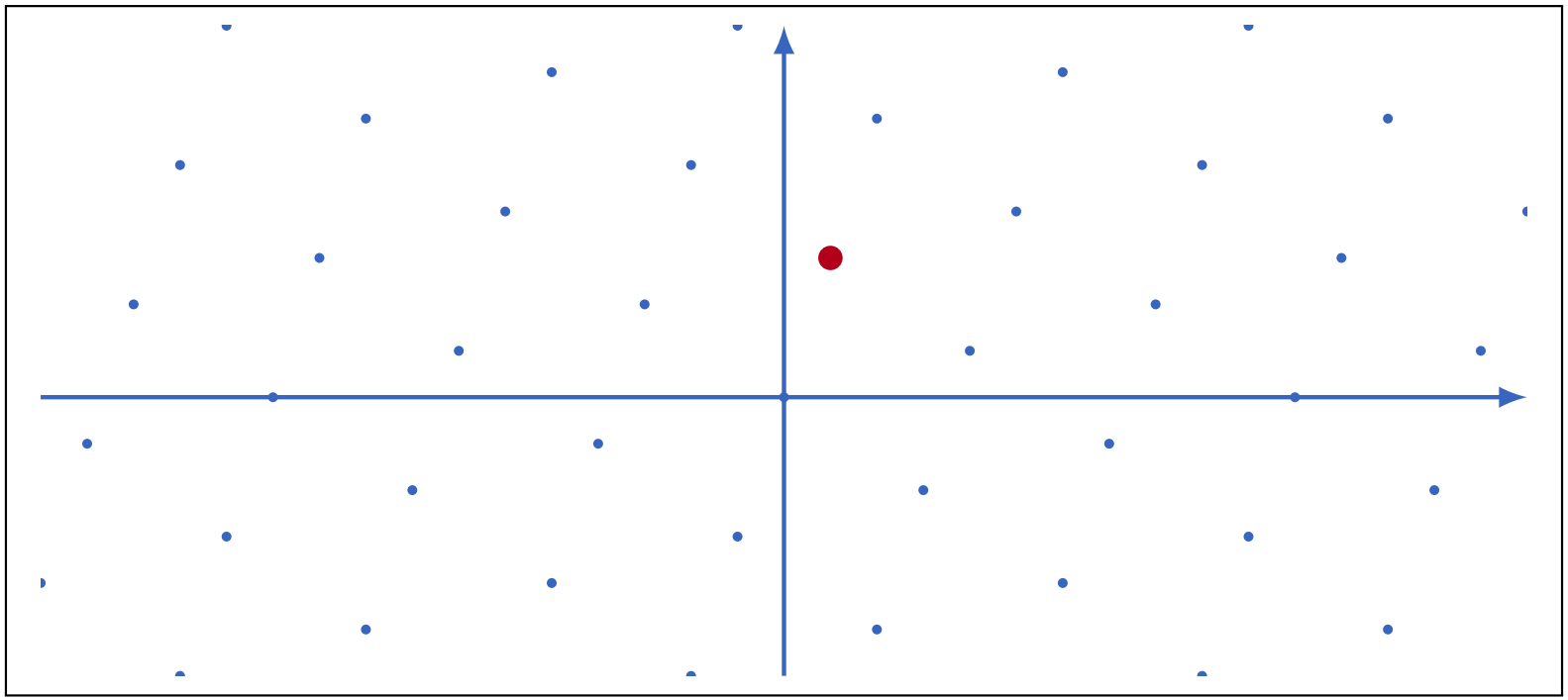


- Any set B of free vectors which generates $\mathcal{L}$ is called a basis.
- There are infinitely many bases.
- Some are better than others : orthogonality, short vectors

Problems on lattices

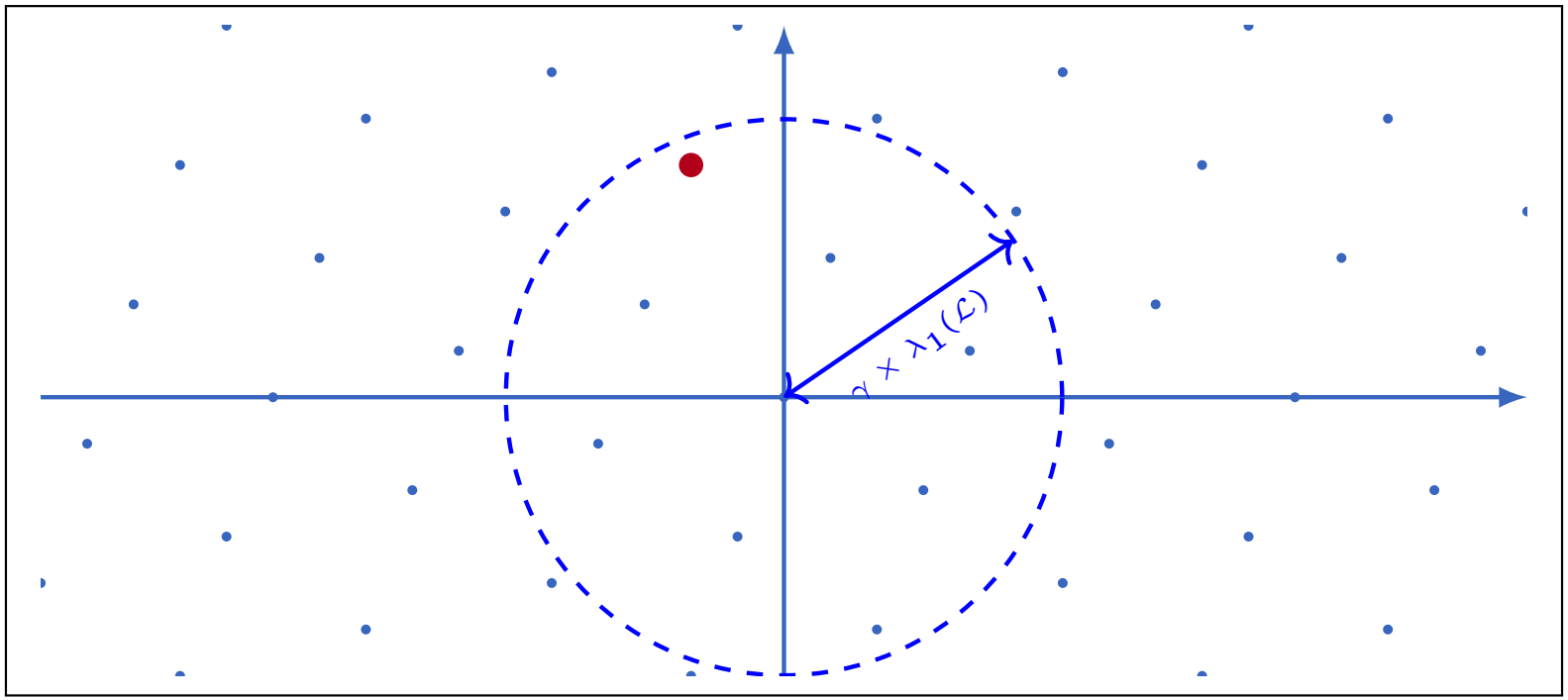


Problems on lattices



Shortest Vector Problem (SVP) : Find a shortest vector of $\mathcal{L} \setminus \{0\}$.
Note $\lambda_1(\mathcal{L})$ its norm.

Problems on lattices



γ-Approximate Shortest Vector Problem (Approx-SVP$_\gamma$, SVP$_\gamma$) : Find a vector of $\mathcal{L}$ with norm less than $\gamma \times \lambda_1(\mathcal{L})$

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Lattice-based cryptography

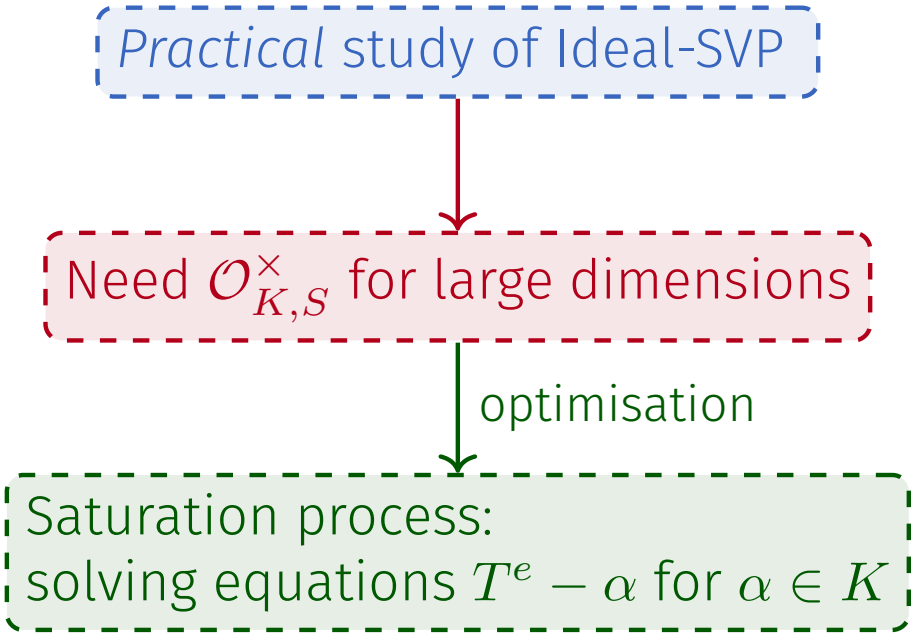
For efficiency reasons, we use structured lattices.

Problems: Ring-LWE or Module-LWE

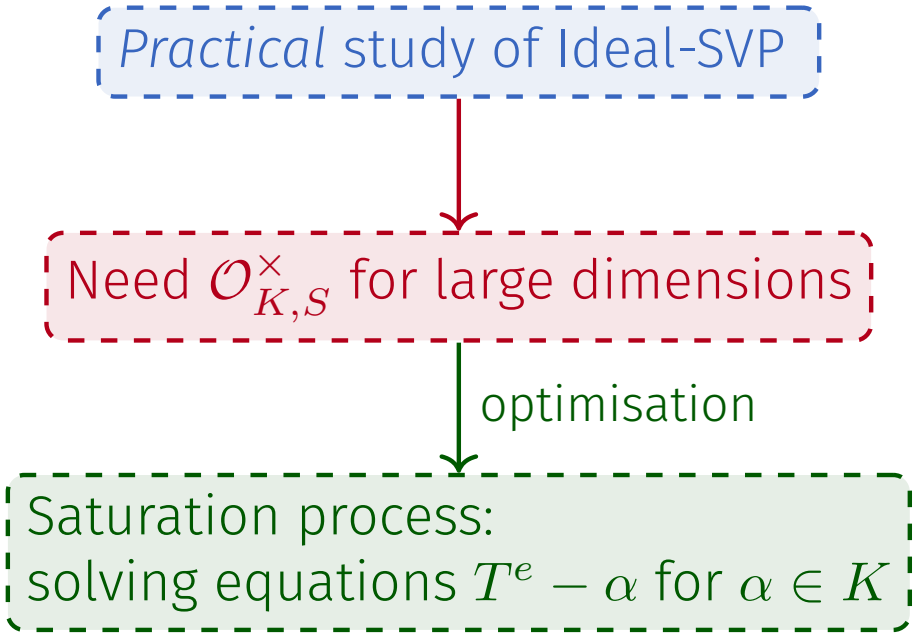
Worst-case to average-cage reductions linking them to SVP over Ideal lattices (Ideal-SVP) or Module lattices (Module-SVP) respectively.

Need to study the hardness of SVP over ideals and modules.

Motivations



Motivations



For today:

The computation of e-th roots for e eventually large. $\rightarrow$ extend Thomé's and Couveignes' methods

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Some definitions

Number field

$$K \cong \mathbb{Q}[X]/(P(X))$$

$$\alpha \in K \iff \text{pol. with coeffs in } \mathbb{Q}$$

$$\mathbb{Q}(\zeta_8) \cong \mathbb{Q}[X]/(X^4+1)$$

$$\alpha = 1/2 + X + 3X^2 - 7/9X^3$$

Some definitions

Number field

$$K \cong \mathbb{Q}[X]/(P(X))$$

$\alpha \in K \iff$ pol. with coeffs in $\mathbb{Q}$

$$\mathbb{Q}(\zeta_8) \cong \mathbb{Q}[X]/(X^4+1)$$

$$\alpha = 1/2 + X + 3X^2 - 7/9X^3$$

Ring of integers

$$\mathcal{O}_K \sim \mathbb{Z}[X]/(P(X))$$

$\alpha \in K \iff$ pol. with coeffs in $\mathbb{Z}$

$$\mathbb{Z}(\zeta_8) \cong \mathbb{Z}[X]/(X^4+1)$$

$$\alpha = 442 - 22X - 519X^2 - 822X^3$$

Recovering elements through a CRT

If α has large coefficients $\Longrightarrow$ compute it modulo several primes.

$$\begin{array}{ccc} \mathcal{O}_K/(p_1 \dots p_r) & \cong & \mathcal{O}_K/(p_1) \times \cdots \times \mathcal{O}_K/(p_r) \\ \alpha & \mapsto & (\alpha \bmod p_1, \dots, \alpha \bmod p_r) \end{array}$$

If $\prod_i p_i > 2\|\alpha\|_\infty$ then α is known :

$$\alpha = 442 - 22X - 519X^2 - 822X^3 \iff \begin{cases} \alpha \bmod 3 = 1 - X \\ \alpha \bmod 17 = -5X + 8X^2 - 6X^3 \\ \alpha \bmod 19 = 5 - 3X - 6X^2 - 5X^3 \end{cases}$$

Recovering elements mod p through a CRT

Consider p a prime integer.

The ideal $p\mathcal{O}_K$ splits into a product of prime ideals : $p\mathcal{O}_K = \prod_{i=1}^{r} \mathfrak{p}_i$.

In $\mathbb{F}_p[X]$: $\bar{P}(X) = \bar{P}_1(X) \times \cdots \times \bar{P}_r(X)$ so we have

$$\begin{array}{ccccc} \mathcal{O}_K(p) & \cong & \mathbb{F}_p[X]/(\bar{P}(X)) & \cong & \mathbb{F}_p[X]/(\bar{P}_1) \times \cdots \times \mathbb{F}_p[X]/(\bar{P}_r) \\ \alpha \bmod p & \mapsto & (\alpha(X) \bmod p) & \mapsto & (\alpha \bmod P_1, \ldots, \alpha \bmod P_r). \end{array}$$

Recovering elements mod p through a CRT

Consider p a prime integer.

The ideal $p\mathcal{O}_K$ splits into a product of prime ideals : $p\mathcal{O}_K = \prod_{i=1}^{r} \mathfrak{p}_i$.

In $\mathbb{F}_p[X]$: $\bar{P}(X) = \bar{P}_1(X) \times \cdots \times \bar{P}_r(X)$ so we have

$$\begin{array}{ccccc} \mathcal{O}_K(p) & \cong & \mathbb{F}_p[X]/(\bar{P}(X)) & \cong & \mathbb{F}_p[X]/(\bar{P}_1) \times \cdots \times \mathbb{F}_p[X]/(\bar{P}_r) \\ \alpha \bmod p & \mapsto & (\alpha(X) \bmod p) & \mapsto & (\alpha \bmod P_1, \ldots, \alpha \bmod P_r). \end{array}$$

If $r = n$, i.e. $\bar{P}(X) = \prod_{i=1}^{n} X - \theta_i$ then $\alpha \bmod P_i = \alpha(\theta_i)$

$$\alpha \bmod 17 = -5X + 8X^2 - 6X^3 \iff \begin{cases} \alpha(1) \bmod 17 = -3 \\ \alpha(8) \bmod 17 = 1 \\ \alpha(9) \bmod 17 = 3 \\ \alpha(15) \bmod 17 = 5 \end{cases}$$

Naive strategy for e-th root computation

Say we know $\beta = \alpha^e$ and we want to compute α .

Compute $\beta_i = \beta \bmod p_i$ for suff. enough p_i

For each i, compute $\alpha_i = \alpha \bmod p_i = \beta_i^{1/e}$

Apply CRT to recover α.

PROBLEM: Potentially e solutions for each $\beta_i^{1/e}(\cdot\zeta_e^j)$

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

A "double CRT" method

Works for fields K s.t. $\exists_\infty$ primes $q \in \mathbb{N}, \forall \mathfrak{q} \mid q, q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e.$ $(*)$

A "double CRT" method

Works for fields K s.t. $\exists_\infty$ primes $q \in \mathbb{N}, \forall \mathfrak{q} \mid q, q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e$. $(*)$

Input : A number field K, $y \in (K^*)^e$

Output : A root $x = y^{1/e}$.

1. Generates suitable primes $q_1, \ldots, q_r$.
2. **For** $i = 1$ **to** r:
   $x_i \leftarrow y^{1/e} \bmod (q_i)$
3. $x \leftarrow \mathrm{CRT}((x_i)_{i \in [\![1,r]\!]}, (q_i)_{i \in [\![1,r]\!]})$.

A "double CRT" method

Works for fields K s.t. $\exists_\infty$ primes $q \in \mathbb{N}, \forall \mathfrak{q} \mid q, q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e$. $(*)$

Input : A number field K, $y \in (K^*)^e$

Output : A root $x = y^{1/e}$.

1. Generates suitable primes $q_1, \ldots, q_r$.
2. **For** $i = 1$ to r:
   $x_i \leftarrow y^{1/e} \bmod (q_i)$
3. $x \leftarrow \mathrm{CRT}((x_i)_{i \in [\![1,r]\!]}, (q_i)_{i \in [\![1,r]\!]})$.

Input : $y \in (K^*)^e$ and a prime $q \in \mathbb{N}$ s.t. $\forall \mathfrak{q} \mid (q)$, $q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e$.

Output : $x \equiv y^{1/e} \bmod (q)$.

1. $S \leftarrow \{\mathfrak{q}, \mathfrak{q} \mid (q)\}$
2. **For** $\mathfrak{q} \in S$:
   $x_\mathfrak{q} \leftarrow y^{1/e} \bmod \mathfrak{q}$
3. $x \leftarrow \mathrm{CRT}((x_\mathfrak{q})_{\mathfrak{q} \in S}, (\mathfrak{q})_{\mathfrak{q} \in S})$

A "double CRT" method

Works for fields K s.t. $\exists_\infty$ primes $q \in \mathbb{N}, \forall \mathfrak{q} \mid q, q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e$. $(*)$

Input : A number field K, $y \in (K^*)^e$

Output : A root $x = y^{1/e}$.

1. Generates suitable primes $q_1, \dots, q_r$.
2. **For** $i = 1$ to r:
   $x_i \leftarrow y^{1/e} \bmod (q_i)$
3. $x \leftarrow \mathrm{CRT}((x_i)_{i \in [\![1,r]\!]}, (q_i)_{i \in [\![1,r]\!]})$.

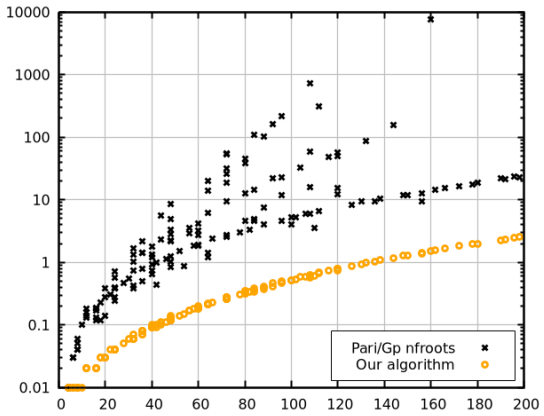
Input : $y \in (K^*)^e$ and a prime $q \in \mathbb{N}$ s.t. $\forall \mathfrak{q} \mid (q)$, $q^{f(\mathfrak{q}|q)} \not\equiv 1 \bmod e$.

Output : $x \equiv y^{1/e} \bmod (q)$.

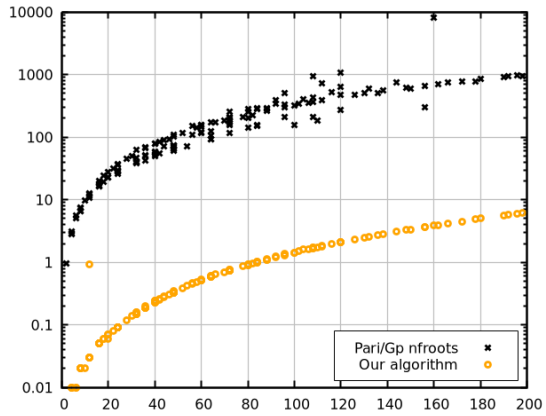
1. $S \leftarrow \{\mathfrak{q}, \mathfrak{q} \mid (q)\}$
2. **For** $\mathfrak{q} \in S$:
   $x_\mathfrak{q} \leftarrow y^{1/e} \bmod \mathfrak{q}$
3. $x \leftarrow \mathrm{CRT}((x_\mathfrak{q})_{\mathfrak{q} \in S}, (\mathfrak{q})_{\mathfrak{q} \in S})$

$(*) \iff$ **There a no ζ_e in any of the residue fields $\mathcal{O}_K/(\mathfrak{q})$**

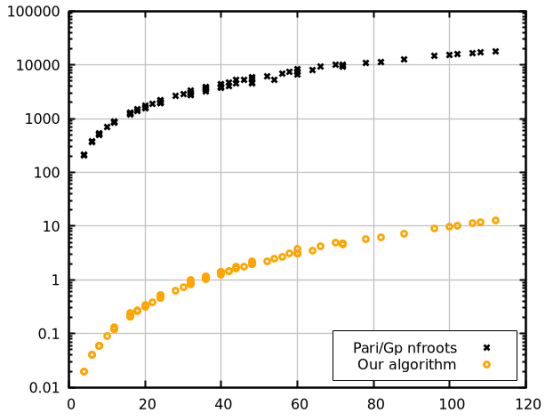
Experimental results



(a) e = 71



(b) e = 1637



(c) e = 13099

Timings (s) for **`nfroots`** and Alg. 2 plotted against the dimension, over cyclotomic fields and $e \in \{71, 1637\}$.

Outline

1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Couveignes' algorithm for square roots

Works for fields K s.t. $[K : \mathbb{Q}]$ is odd and $\exists_\infty$ inert primes $p \in \mathbb{N}$.

1. Select inert primes $p_1, \ldots, p_n$
2. For each p_i, compute $x_i = \sqrt{y} \bmod p_i$
3. Output $x = \mathrm{CRT}((x_i)_{i \in [\![1,n]\!]}, (p)_{i \in [\![1,n]\!]})$

Couveignes' algorithm for square roots

Works for fields K s.t. $[K:\mathbb{Q}]$ is odd and $\exists_\infty$ inert primes $p \in \mathbb{N}$.

1. Select inert primes $p_1, \ldots, p_n$
2. For each p_i, compute $x_i = \sqrt{y} \bmod p_i$
3. Output $x = \mathsf{CRT}((x_i)_{i\in[\![1,n]\!]}, (p)_{i\in[\![1,n]\!]})$

How to select between $\pm x_i$?

We have a commutative diagram :

$$\begin{array}{ccc} \mathcal{O}_K & \twoheadrightarrow & \mathcal{O}_K/(p_i) \\ {\scriptstyle \mathbf{N}}\downarrow & \circlearrowleft & \downarrow{\scriptstyle \mathbf{N}} \\ \mathbb{Z} & \twoheadrightarrow & \mathbb{F}_{p_i} \end{array}$$

Remark that $\mathbf{N}(\pm x) = \pm \mathbf{N}(x)$ so:

- ⋆ Fix $\mathbf{N}(x) := \sqrt{\mathbf{N}(y)}$
- ⋆ For each i, choose the root s.t. $\mathbf{N}(x_i \bmod p_i) = \mathbf{N}(x) \bmod p_i$

Couveignes' algorithm for larger e

Works for extensions K/k s.t. $\gcd([K:k], e) = 1$, $\zeta_e \in k$ and $\exists_\infty p \in \mathbb{N}$ prime with $\forall \mathfrak{p} \mid p$ is inert in K/k

1. Select $p_1, \ldots, p_r \in \mathbb{N}$ such that $\forall \mathfrak{p} \mid p_i$ in k, $\mathfrak{p}$ is inert in K/k
2. For each p_i, compute $x_i = y^{1/e} \bmod p_i$
3. Output $x = \mathrm{CRT}((x_i)_{i \in [\![1,r]\!]}, (p_i)_{i \in [\![1,r]\!]})$

Couveignes' algorithm for larger e

Works for extensions K/k s.t. $\gcd([K:k], e) = 1$, $\zeta_e \in k$ and $\exists_\infty p \in \mathbb{N}$ prime with $\forall \mathfrak{p} \mid p$ is inert in K/k

1. Select $p_1, \ldots, p_r \in \mathbb{N}$ such that $\forall \mathfrak{p} \mid p_i$ in k, $\mathfrak{p}$ is inert in K/k
2. For each p_i, compute $x_i = y^{1/e} \bmod p_i$
3. Output $x = \mathrm{CRT}((x_i)_{i \in [\![1,r]\!]}, (p_i)_{i \in [\![1,r]\!]})$

How to compute all x_i in a coherent way? $\rightarrow$ use a CRT à la Couveignes

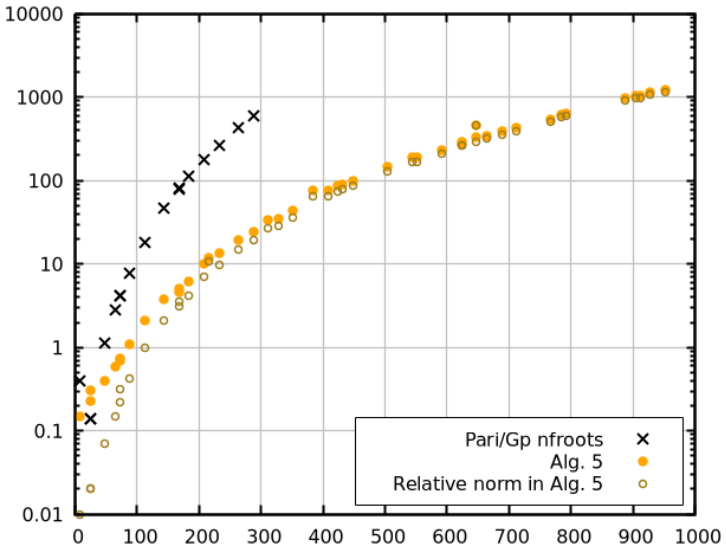
For one p_i and one $\mathfrak{p} \mid p_i$:

$$\begin{array}{ccc} \mathcal{O}_K & \twoheadrightarrow & \mathcal{O}_K/\mathfrak{p} \\ {\scriptstyle \mathbf{N}}\downarrow & \circlearrowleft & \downarrow{\scriptstyle \mathbf{N}} \\ \mathcal{O}_k & \twoheadrightarrow & \mathcal{O}_k/\mathfrak{p} \end{array}$$

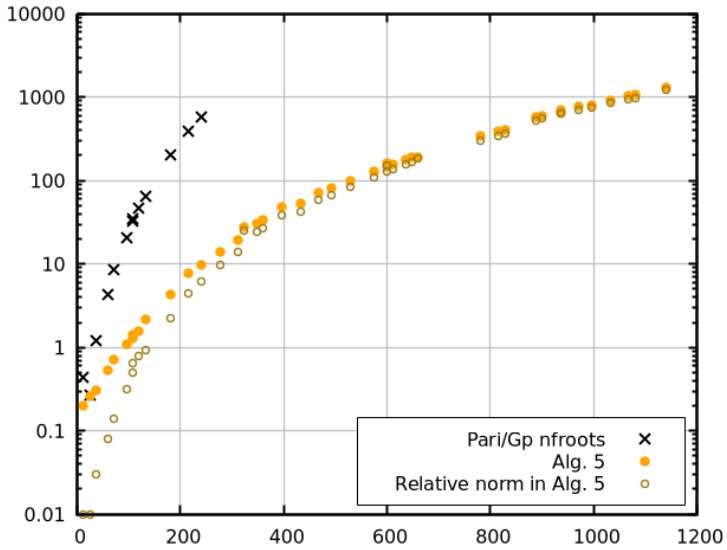
$\{\mathbf{N}(\zeta_e^i x), 0 \leqslant i < e\} = \{\zeta_e^j \mathbf{N}(x), 0 \leqslant j < e\}$ so:

- ⋆ Fix $\mathbf{N}(x) := \mathbf{N}(y)^{1/e}$ e-th root in a subfield
- ⋆ For each i and each $\mathfrak{p} \mid p_i$, choose the root s.t. $\mathbf{N}(x_i \bmod \mathfrak{p}) = \mathbf{N}(x) \bmod \mathfrak{p}$

Experimental results



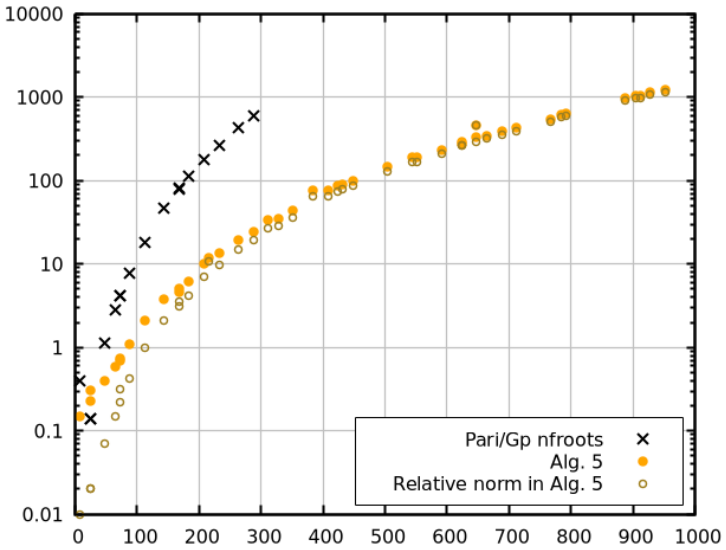
(a) e = 5



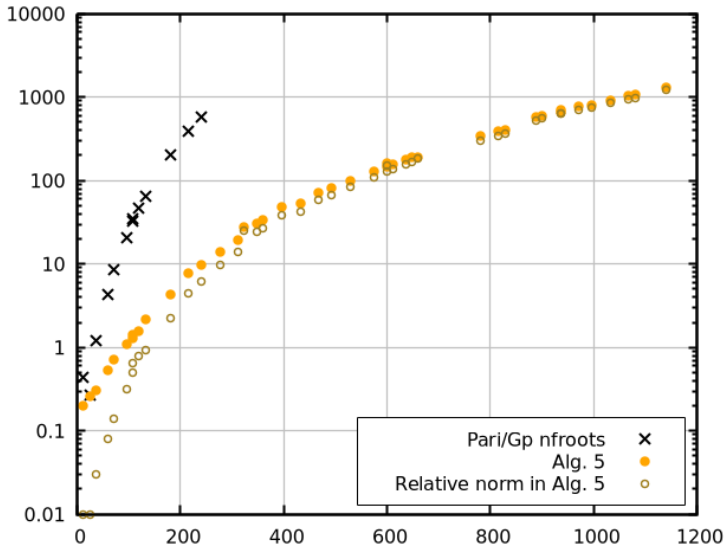
(b) e = 7

Timings (s) for **`nfroots`** and Generalised Couveignes plotted against the dimension, over cyclotomic fields and $e \in \{3, 5\}$.

Experimental results



(a) e = 5



(b) e = 7

Timings (s) for **`nfroots`** and Generalised Couveignes plotted against the dimension, over cyclotomic fields and $e \in \{3, 5\}$.

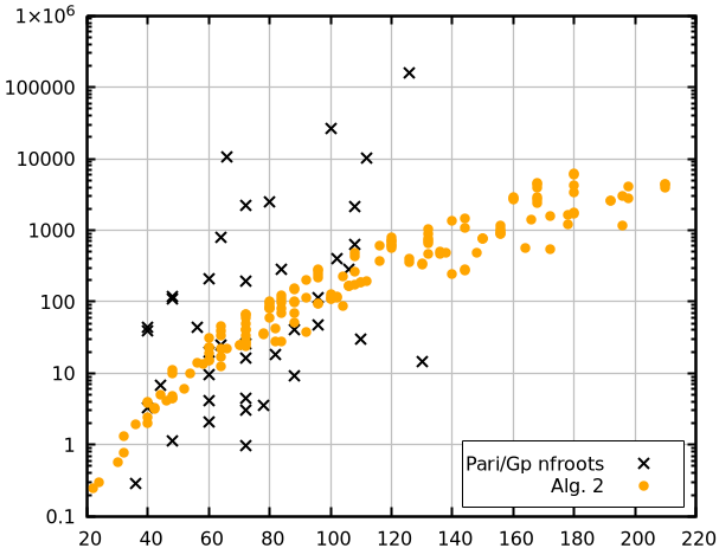
Now the main obstruction is the computation of $\mathrm{N}_{K/k}(y)$

Outline

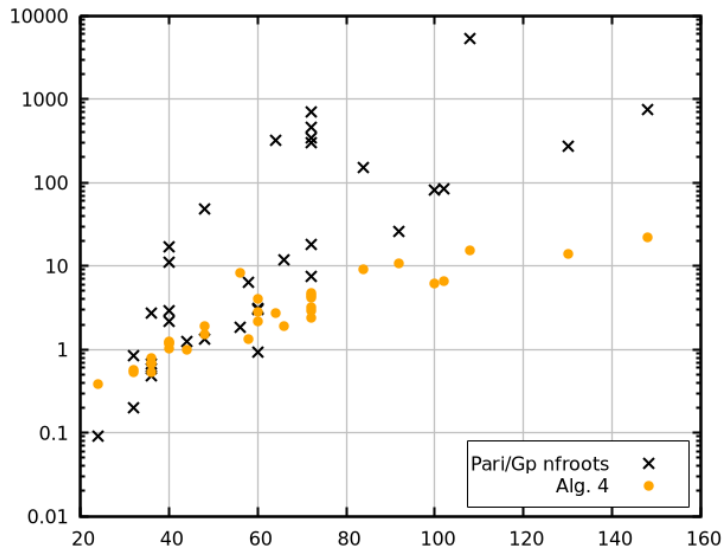
1. Context
   - Lattice-based cryptography
   - (Structured) Lattices
   - Background on number fields

2. Computing n-th roots in number fields
   - Easy cases: generalisation of Thomé's algorithm
   - Hard cases: generalisation of Couveignes' algorithm
   - Experiments for saturation

Some figures



(a) $e \not\mid m$



(b) $e \mid m$

Timings (s) for **`nfroots`** and our algorithms over cyclotomic fields within computation of S-units.

More timings

Table: Timings (s) for e-th roots within computation of S-units for selected cyclotomic fields

Conductor m	53	61	67	107	109	151	199
Dim. $\varphi(m)$	52	60	66	106	108	150	198
e ($\mathbf{log_2}(e)$)	4889 (12)	1861 (10)	12739 (13)	2886593 (21)	9431866153 (33)	312885301(28)	207293548177 (37)
Our algorithm	6.08	15.16	22.51	166.12	207.89	775.02	3153.98
nfroots	738.98	210.98	10528.81	n/a	n/a	n/a	n/a

Thank you for your attention