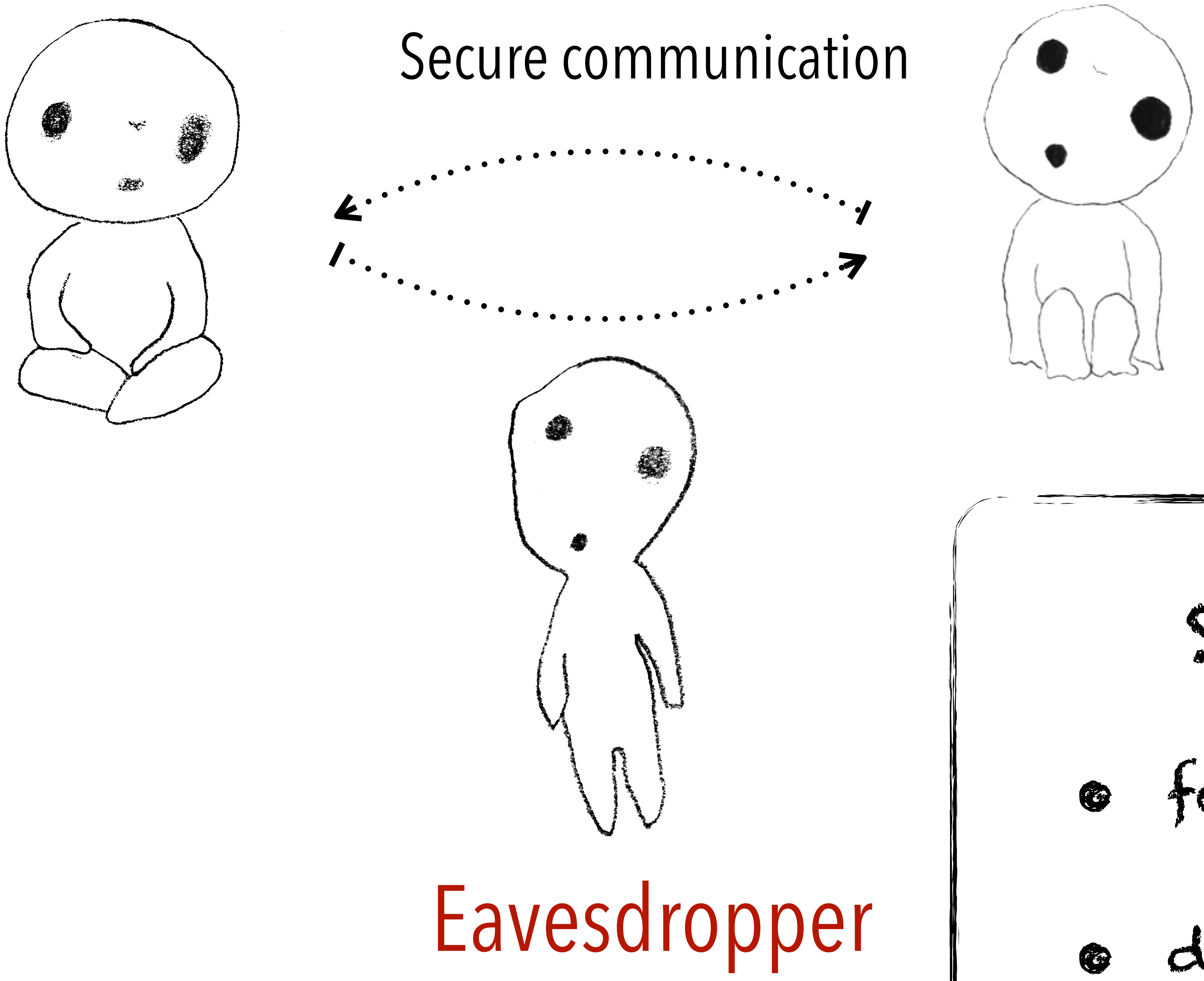
Cryptanalyses de logarithmes discrets

Journées de la sécurité GDR 2022

Gabrielle De Micheli
Université de Californie, San Diego

Joint work with Pierrick Gaudry and Cécile Pierrot

Hard problems for Cryptography



Use (hopefully) intractable problems to construct cryptographic primitives.

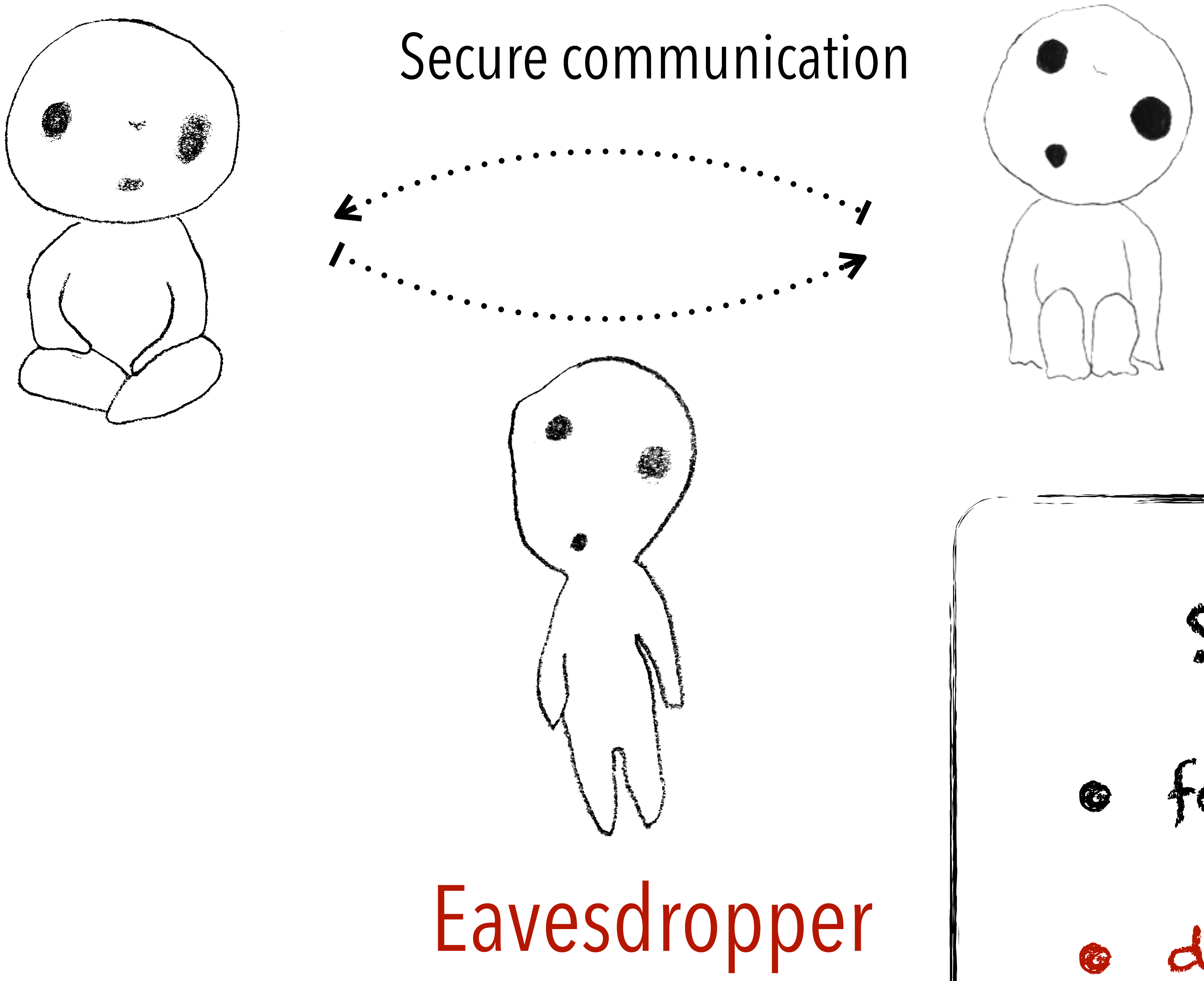
Start from...

- factorisation
- discrete logarithm
- lattice problems
- isogeny problems
- ...

... to obtain:

- encryption schemes
- signature schemes
- hash functions
- ...

Hard problems for Cryptography



Use (hopefully) intractable problems to construct cryptographic primitives.

Start from...

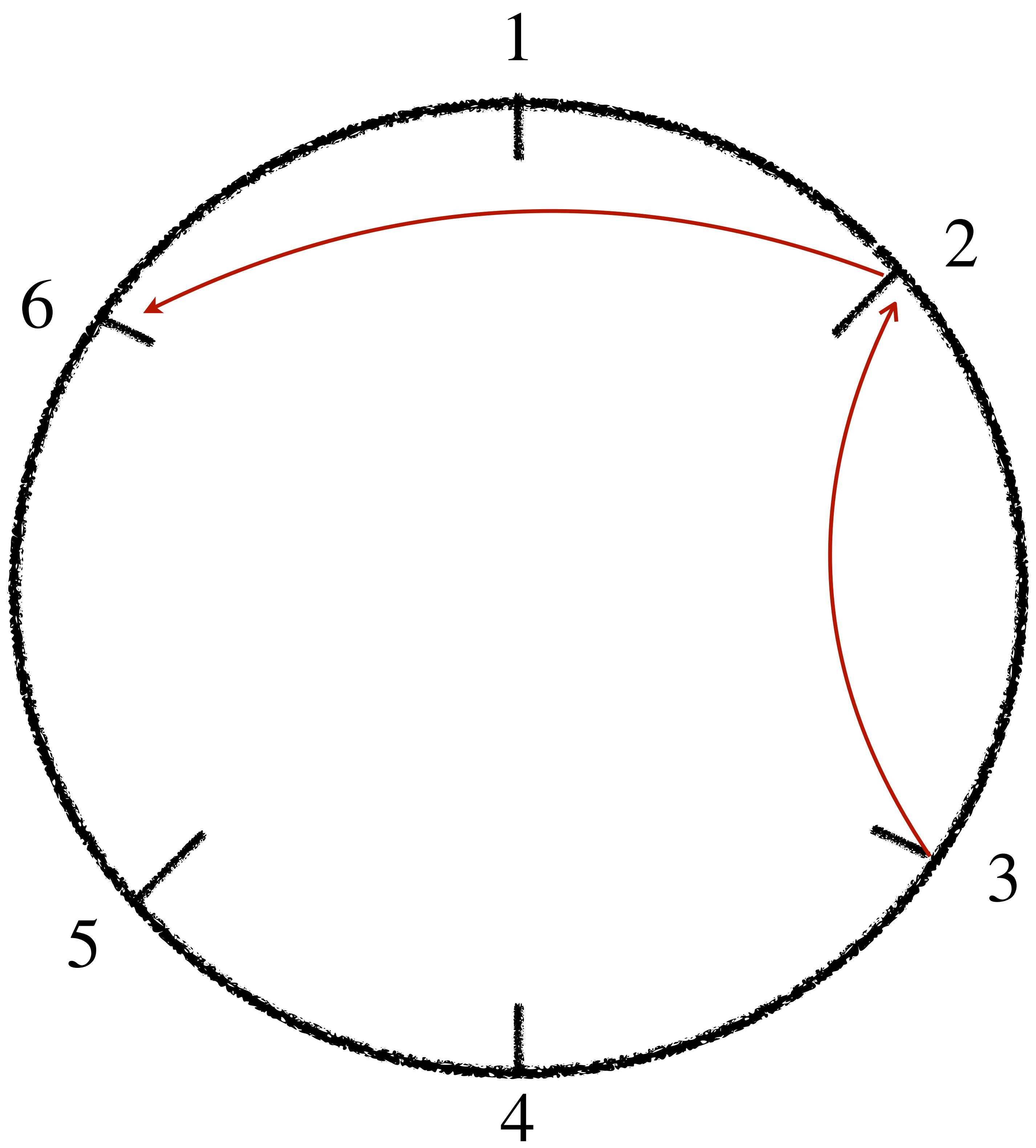
- factorisation
- discrete logarithm
- lattice problems
- isogeny problems
- ...

... to obtain:

- encryption schemes
- signature schemes
- hash functions
- ...

What is a discrete logarithm?

Definition: Given a finite cyclic group G of order n, a generator $g \in G$ and some element $h \in G$, the discrete logarithm of h in base g is the element $x \in [0,n)$ such that $g^x = h$.



Example: $G = \mathbb{Z}_7^\times,\ g = 3,$
$h = 6 \in \mathbb{Z}_7^\times,$

$$g^1 \equiv 3 \pmod 7$$
$$g^2 = 9 \equiv 2 \pmod 7$$
$$g^3 = 27 \equiv 6 \pmod 7$$

The discrete logarithm of h in base g is 3.

The discrete logarithm problem (DLP)

Definition: Given a finite cyclic group G of order n, a generator $g \in G$ and some element $h \in G$, **find** the element $x \in [0,n)$ such that $g^x = h$.

Computing the inverse, a modular exponentiation is easy:

$$g^x = \underbrace{g \cdot g \cdot \cdots \cdot g}_{x}$$

algorithms in $O(\log(x))$

Solving DLP can be hard (depending on the group G):

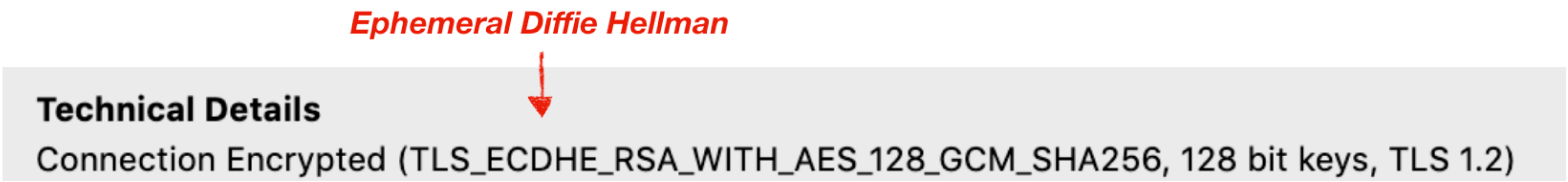
$$h = \underbrace{g \cdot g \cdot \cdots \cdot g}_{??}$$

Why do we care about discrete logarithms?

Many protocols use modular exponentiation where the exponent is a secret.

Example 1: Diffie-Hellman key exchange [DH76]

- Public data: $g, g^a, g^b \in G$
- Shared key: $g^{ab} \in G$



Example 2: pairing-based protocols

- Identity-based encryption/signature schemes [BF01], [CC03]
- Short signature schemes (eg, BLS signatures [BLS01])

Security based on assumptions that become false if DLP is broken.

[DH76]: W. Diffie, M. Hellman, New directions in cryptography. Trans. Info. Theory, 1976

[BF01]: D. Boneh, M. Franklin, Identity-based encryption from Weil pairing. Crypto'01

[CC03]: J. Cha, J. Cheon, An identity-based signature from gap Diffie-Hellman groups. PKC'03

[BLS01]: D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing. Asiacrypt'01

In my work

How can we assess the security of protocols in which a modular exponentiation involving a secret exponent is performed?

- Estimate the hardness of DLP in the groups considered by the protocols.
- Look at implementation vulnerabilities during fast exponentiation.

An example: EPID protocol in Intel SGX

- What is EPID? a protocol to allow remote attestation of a hardware platform without compromising the device's identity.

- The protocol includes a signing algorithm that uses pairings.

  - secret key includes the element $f \in_R \mathbb{Z}_q$

- How can we recover f?

  - During the protocol, consider a random secret nonce $r \in \mathbb{Z}_q$

  - Compute an exponentiation X^r

  - Outputs the element $s \leftarrow r + cf$ ($c =$ hash of known values)

How can we recover the secret f ?

Since $s \leftarrow r + cf$, if we recover r , we directly get f.

The protocol uses a 256-bit elliptic curve Fp256BN (embedding degree 12).

If we have as target X^r :

1. Solve DLP to find exponent r in 3072-bit finite field $\mathbb{F}_{p^{12}}$.

2. Look at implementation vulnerabilities during the computation of X^r.

In my work

How can we assess the security of protocols in which a modular exponentiation involving a secret exponent is performed?

- Estimate the hardness of DLP in the groups considered by the protocols.
- ~~Look at implementation vulnerabilities during fast exponentiation.~~

The discrete logarithm problem over finite fields

Definition: Given a finite cyclic group G of order n, a generator $g \in G$ and some element $h \in G$, find the element $x \in [0,n)$ such that $g^x = h$.

What group G should be considered?

$$(\cancel{\mathbb{Z}/n\mathbb{Z}}, +)$$

- Prime finite fields $\mathbb{F}_p^\times$
- Finite fields $\mathbb{F}_{p^n}^\times$
- Elliptic curves over finite fields $\mathcal{E}(\mathbb{F}_p)$
- Genus 2 hyperelliptic curves

The discrete logarithm problem over finite fields

Definition: Given a finite cyclic group G of order n, a generator $g \in G$ and some element $h \in G$, find the element $x \in [0,n)$ such that $g^x = h$.

What group G should be considered?

$$\cancel{(\mathbb{Z}/n\mathbb{Z}, +)}$$

- Prime finite fields $\mathbb{F}_p^\times$
- Finite fields $\mathbb{F}_{p^n}^\times$
- Elliptic curves over finite fields $\mathscr{E}(\mathbb{F}_p)$
- Genus 2 hyperelliptic curves

Evaluating the hardness of DLP over $\mathbb{F}_{p^n}$

- Many different algorithms to solve DLP in $\mathbb{F}_{p^n}$.
- Their complexities depend on the relation between the characteristic p and the extension degree n.

A useful notation: the L-notation $$\boxed{L_{p^n}(\alpha, c) = \exp((c + o(1))\log(p^n)^\alpha \log\log(p^n)^{1-\alpha})}$$

for $0 \leqslant \alpha \leqslant 1$ and $c > 0$.

For complexities:

- When $\alpha \to 0$: $\exp(c \log\log p^n) \approx (\log p^n)^c$, polynomial-time
- When $\alpha \to 1$: p^{cn}, exponential-time

In the <u>middle</u>: subexponential-time

Three families of finite fields

Finite field $\mathbb{F}_{p^n}$ with $p = L_{p^n}(\alpha, c)$

- Different algorithms are used in the different areas.
- Algorithms don't have the same complexity in each area.

Index calculus algorithms

Consider a finite field $\mathbb{F}_{p^n}$

Factor basis: $\mathscr{F} =$ small set of small elements

Three main steps:

- Relation collection: find relations between the elements of $\mathscr{F}$.
- Linear algebra: solve a system of linear equations where the unknowns are the discrete logarithms of the elements of $\mathscr{F}$.
- Individual logarithm/Descent: for a target element $h \in \mathbb{F}_{p^n}^{\times}$, compute the discrete logarithm of h.

A lot of algorithms

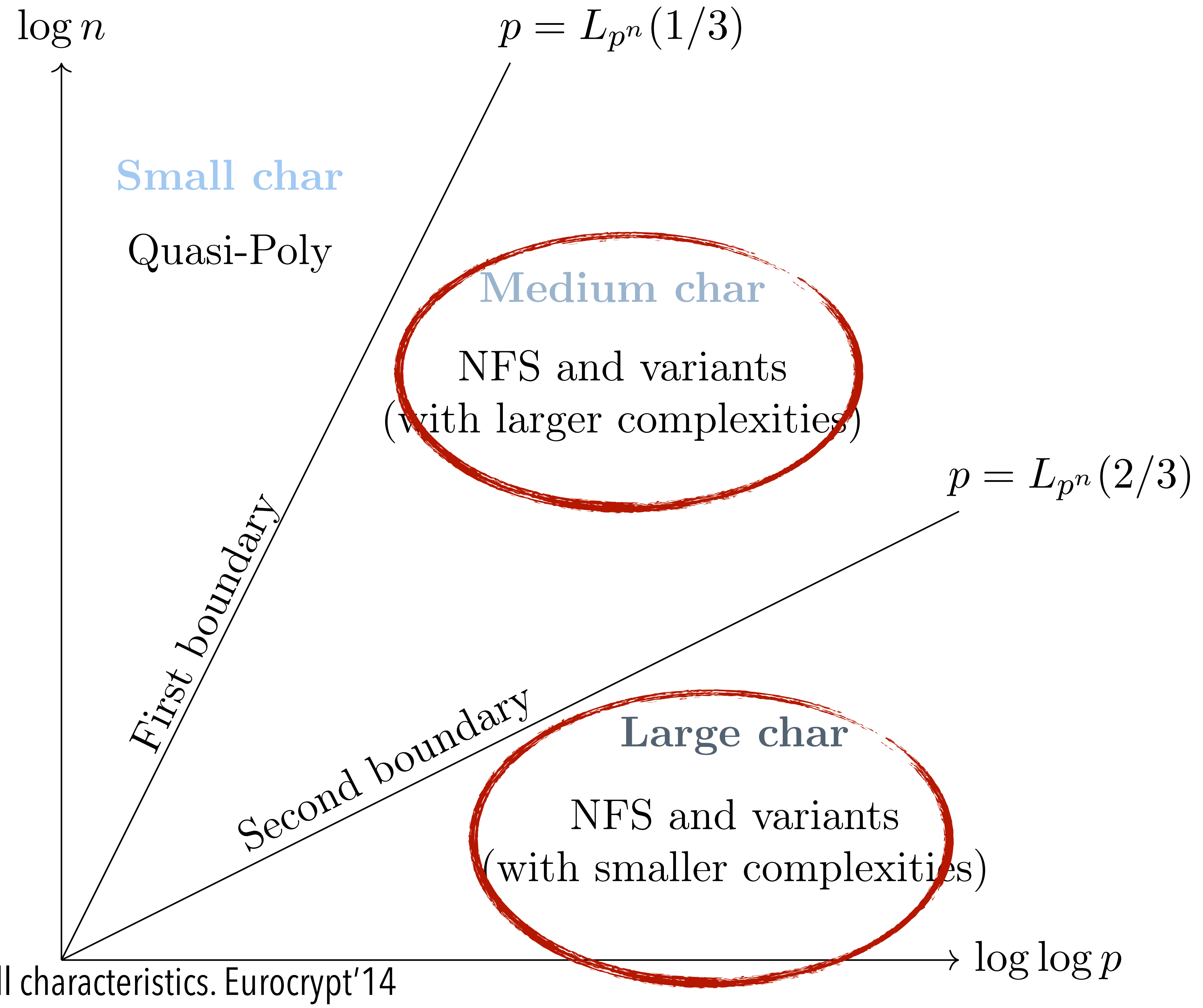
- Small characteristics: Quasi-Polynomial algorithms [BGJT14, KW19] (with only a descent step) and Function Field Sieve [Adl94]

- Medium and large characteristics: Number Field Sieve (NFS) [Gor93] and its variants

We focus on medium and large characteristic finite fields.

Why?

Finite fields used in practice for example $\mathbb{F}_{p^6}$ for MNT-6 elliptic curves in zk-SNARKS.



[Adl94]: L. Adleman, The Function Field Sieve. ANTS'94
[Gor98]: D. Gordon, Discrete Logarithms in GF(P) Using the Number Field Sieve. Journal on Discrete Mathematics'93
[BGJT14]: R. Barbulescu, P. Gaudry, A. Joux, E. Thomé, A heuristic quasi-polynomial time algorithm for discrete logarithm in finite fields of small characteristics. Eurocrypt'14
[KW19]: T. Kleinjung, B. Wesolowski, Discrete logarithms in quasi-polynomial time in finite fields of fixed characteristic. 2019

Why do we do record computations?

It is important to choose the right key size.

- Too large: needlessly expensive computations
- Too small: insecure

Agency	Date	Size of group	Size of key
NIST	2019-2030	2048	224
	> 2030	3072	256
ANSSI	2021-2030	2048	200
	> 2030	3072	200

Running-time of discrete logarithm algorithms is hard to predict.

Record computations provide information for assessing key lifetime.

A first record computation with exTNFS

- Why did we choose exTNFS?

$$n = \eta\kappa$$

$$\mathbb{F}_{p^n} = \mathbb{F}_{p^{\eta\kappa}} = \mathbb{F}_{P^\kappa}$$

Specificity	Algorithm	Medium characteristic	2nd boundary	Large characteristic
None	NFS	96	48	64
	MNFS	89.45	45.00	61.93
	TNFS	–	–	64
	MTNFS	–	–	61.93
Composite n	exTNFS	48	–	–
	MexTNFS	45.00	–	–
Special p	SNFS	$64\left(\frac{\lambda+1}{\lambda}\right)$	$\star$	32
	STNFS	–	–	32
Composite n and special p	SexTNFS	32	$\star$	32

- Main difficulty: relation collection in dimension > 2.

[KB16]: T. Kim, R. Barbulescu, Extended tower number field sieve. Crypto'16

Collecting relations in TNFS

- Relation collection: find relations between the elements of $\mathscr{F}$.

More precisely, what does this mean? What is a relation? Who is $\mathscr{F}$?

For TNFS: $R = \mathbb{Z}[\iota]/h(\iota)$

In our computation:

- $n = 6 = 3 \times 2$
- $\deg h = \eta = 3$
- $h = \iota^3 - \iota + 1$

$$\begin{tikzcd}
& R[X] \arrow[dl] \arrow[dr] & \\
K_1 \supset R[X]/(X^4+1) \arrow[dr] & & K_2 \supset R[X]/(aX^2+bX+c) \arrow[dl] \\
& (R/pR)[X]/(aX^2+bX+c) \cong \mathbb{F}_{p^6} &
\end{tikzcd}$$

Collecting relations in TNFS

- Relation collection: find relations between the elements of $\mathscr{F}$.

For TNFS: $R = \mathbb{Z}[\iota]/h(\iota)$

In our computation:

- $n = 6 = 3 \times 2$
- $\deg h = \eta = 3$
- $h = \iota^3 - \iota + 1$

More precisely, what does this mean? What is a relation? Who is $\mathscr{F}$?

$$\begin{tikzcd}
& R[X] \arrow[dl] \arrow[dr] \arrow[dd, bend right] \arrow[dd, bend left] & \\
K_1 \supset R[X]/(X^4+1) \arrow[dr] & & K_2 \supset R[X]/(aX^2+bX+c) \arrow[dl] \\
& \mathbb{F}_{p^6} &
\end{tikzcd}$$

Collecting relations in TNFS: what is a relation?

$$R = \mathbb{Z}[\imath]/(\imath^3 - \imath + 1)$$

$$\begin{tikzcd}
 & \phi(\imath, X) = a(\imath) - b(\imath)X \in R[X] \arrow[dl] \arrow[dr] & \\
K_1 \supset R[X]/(X^4+1) \arrow[dr] & & K_2 \supset R[X]/(aX^2+bX+c) \arrow[dl] \\
 & \mathbb{F}_{p^6} &
\end{tikzcd}$$

$$\phi(\imath, \alpha_1) = a(\imath) - b(\imath)\alpha_1$$

Test $N(\phi(\imath, \alpha_1))$ for B-smoothness:

$\longrightarrow$ prime factors smaller than B

Equality in finite field = Relation

Collecting relations in TNFS: what is a relation?

- Relation collection: find relations between the elements of $\mathscr{F}$.

$$\begin{tikzcd} & R[X] \arrow[dl] \arrow[dr] & \\ K_1 \supset R[X]/(X^4+1) \arrow[dr] & & K_2 \supset R[X]/(aX^2+bX+c) \arrow[dl] \\ & \mathbb{F}_{p^6} & \end{tikzcd}$$

Who is $\mathscr{F}$?

Prime ideals of small norm in the ring of integers of the intermediate number fields

$$\prod \mathfrak{p}_i^{e_i} \text{ " = " } \prod \mathfrak{q}_j^{f_j}$$

Collecting relations in TNFS

Relation collection looks for a set of linear polynomials $\phi(\iota, X) = a(\iota) - b(\iota)X \in R[X]$

1. with bounded coefficients → $c \in \mathcal{S}$ where $\mathcal{S}$ is known as the sieving region.

2. such that $N_i(a(\iota) - b(\iota)\alpha_i)$ is B-smooth → Norms divisible only by primes smaller than B:
$c \in$ intersection of suitably constructed lattices $\mathcal{L}$

Concretely, let:

$$a(\iota) = a_0 + a_1\iota + a_2\iota^2$$

$$b(\iota) = b_0 + b_1\iota + b_2\iota^2$$

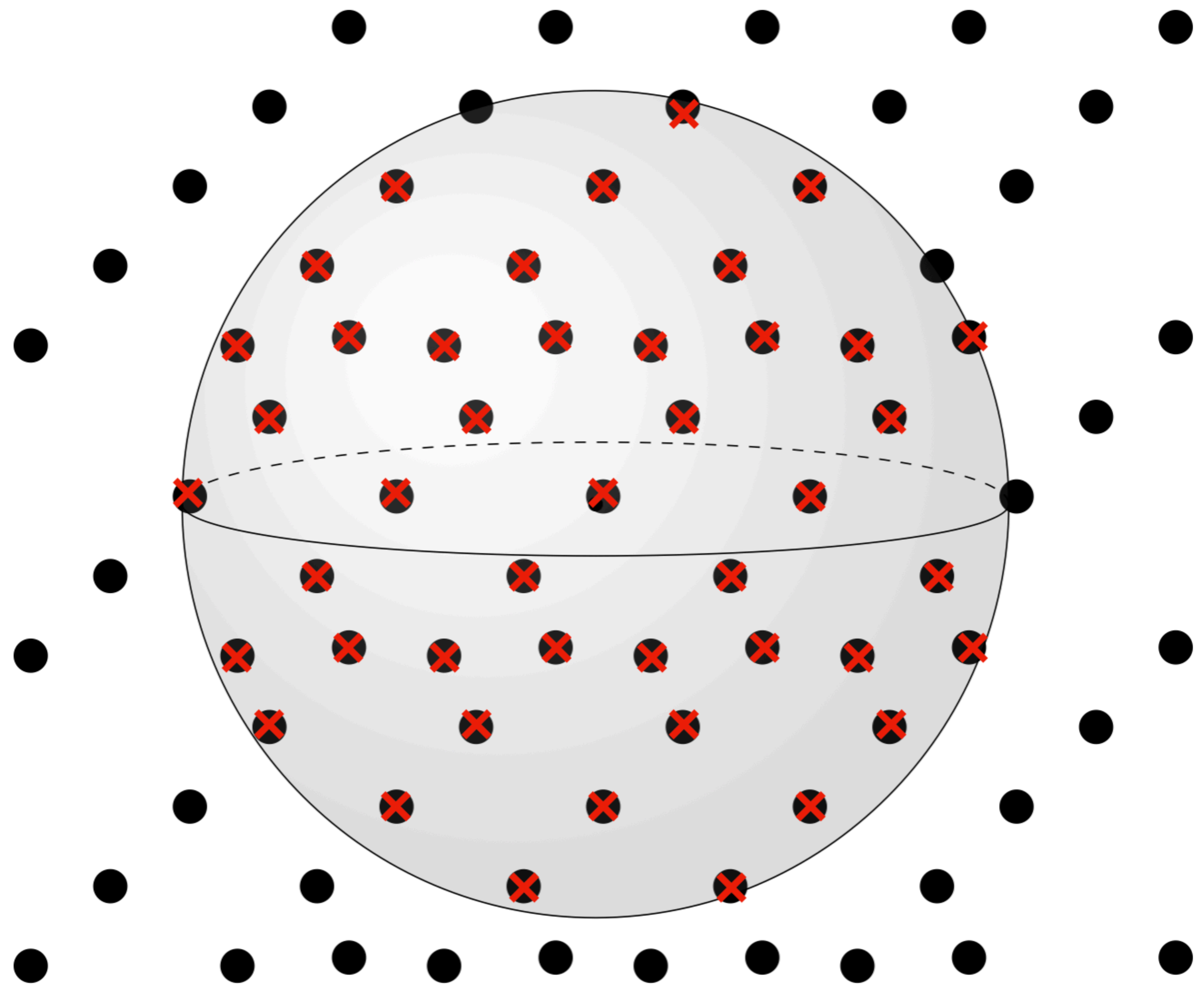
Goal: find vectors $c = (a_0, a_1, a_2, b_0, b_1, b_2) \in \mathbb{Z}^6$ such that

A new sieving region

Goal: find $c \in \mathcal{S} \cap \mathcal{L}$

What is the dimension of $\mathcal{S}$? $d = 2\eta = 6$

We look at TNFS so dimension > 2 (since $\eta \geq 2$) and $\mathcal{S} = 6$-**sphere** (**ℓ_2-norm).**



Enumerating in $\mathscr{S} \cap \mathscr{L}$

- Concretely what is $\mathscr{L}$?

A lattice that describes the divisibility of the ideals by an ideal $\mathfrak{Q}$, known as a special-q ideal and a prime ideal $\mathfrak{p}$ in the intermediate number fields.

for many $\mathfrak{p}'s$

- The outputs of the enumeration are thus …

…vectors corresponding to (a, b) pairs whose norms are divisible by $N(\mathfrak{Q})$ and $N(\mathfrak{p})$.

Why? high probability of B-smoothness

Schnorr-Euchner's enumeration [SE94]

- Input: a lattice basis $\mathbf{b}_1, \cdots, \mathbf{b}_d$
- Output: shortest non-zero lattice vector

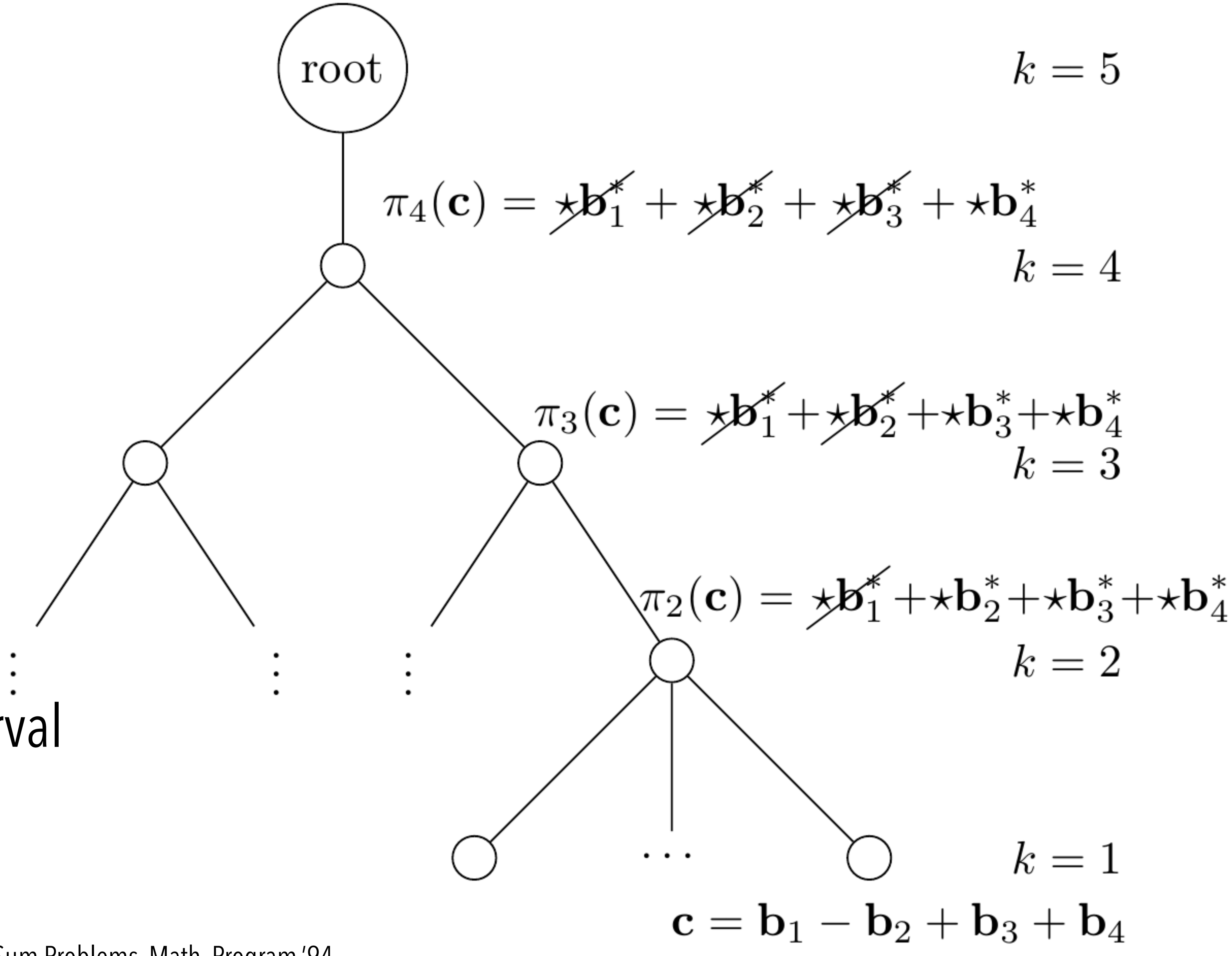
Idea:

1. Construct an enumeration tree

2. Consider projections of the lattice

3. At each level of the tree, enumerate in an interval

4. Depth-first search in the tree



[SE94]: C-P. Schnorr, M. Euchner, Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. Math. Program.'94

Schnorr-Euchner's enumeration [SE94]

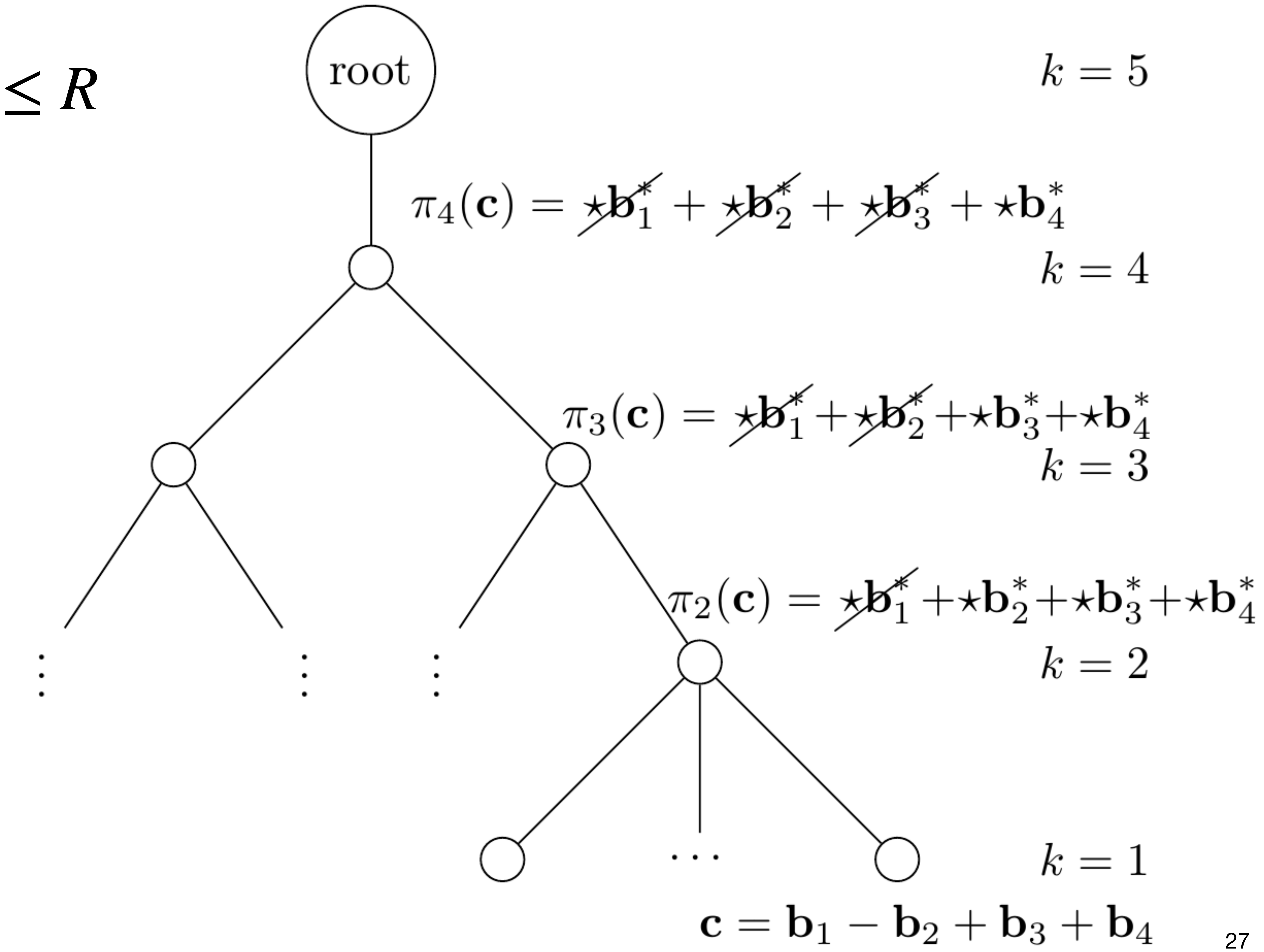
- Input: a lattice basis $\mathbf{b}_1, \cdots, \mathbf{b}_6$
- Output: vectors $c = \sum v_i \mathbf{b}_i$ such that $||c|| \leq R$

Idea:

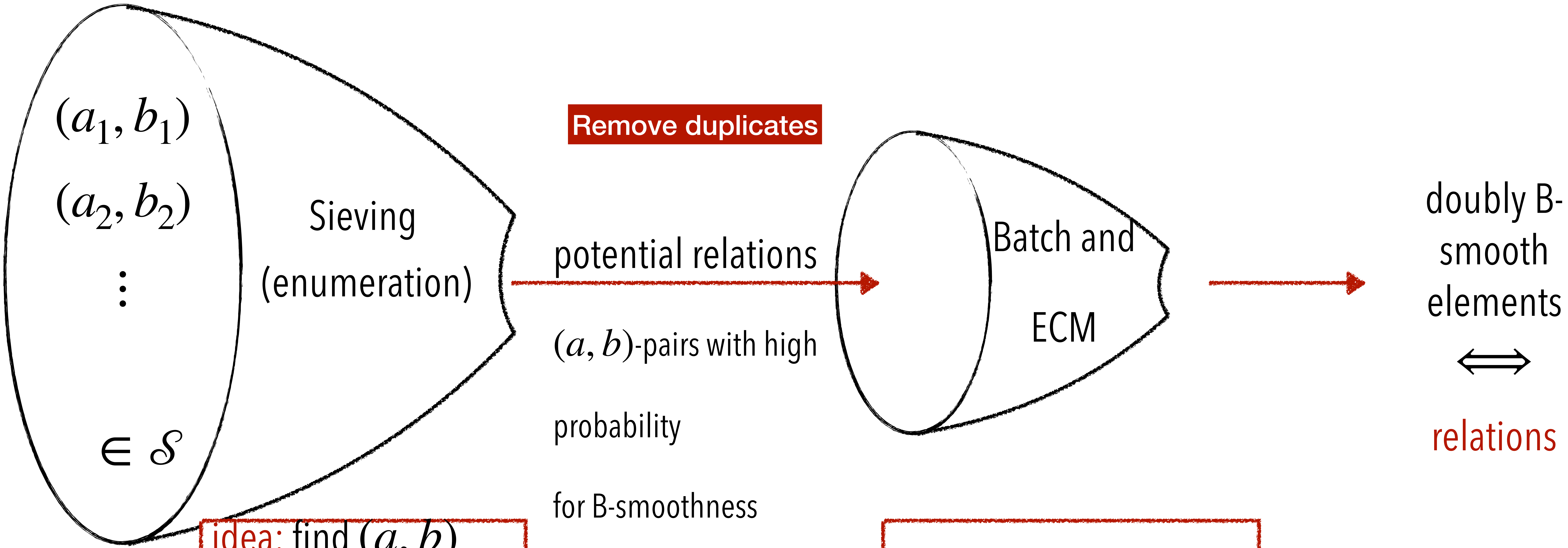
1. Construct an enumeration tree

2. Consider projections of the lattice

3. Exhaustive search of the coefficients v_i



Relation collection all together



idea: find (a, b) -pairs in $\mathcal{S}$ intersect many lattices $\mathcal{L}_{\mathfrak{p}}$.

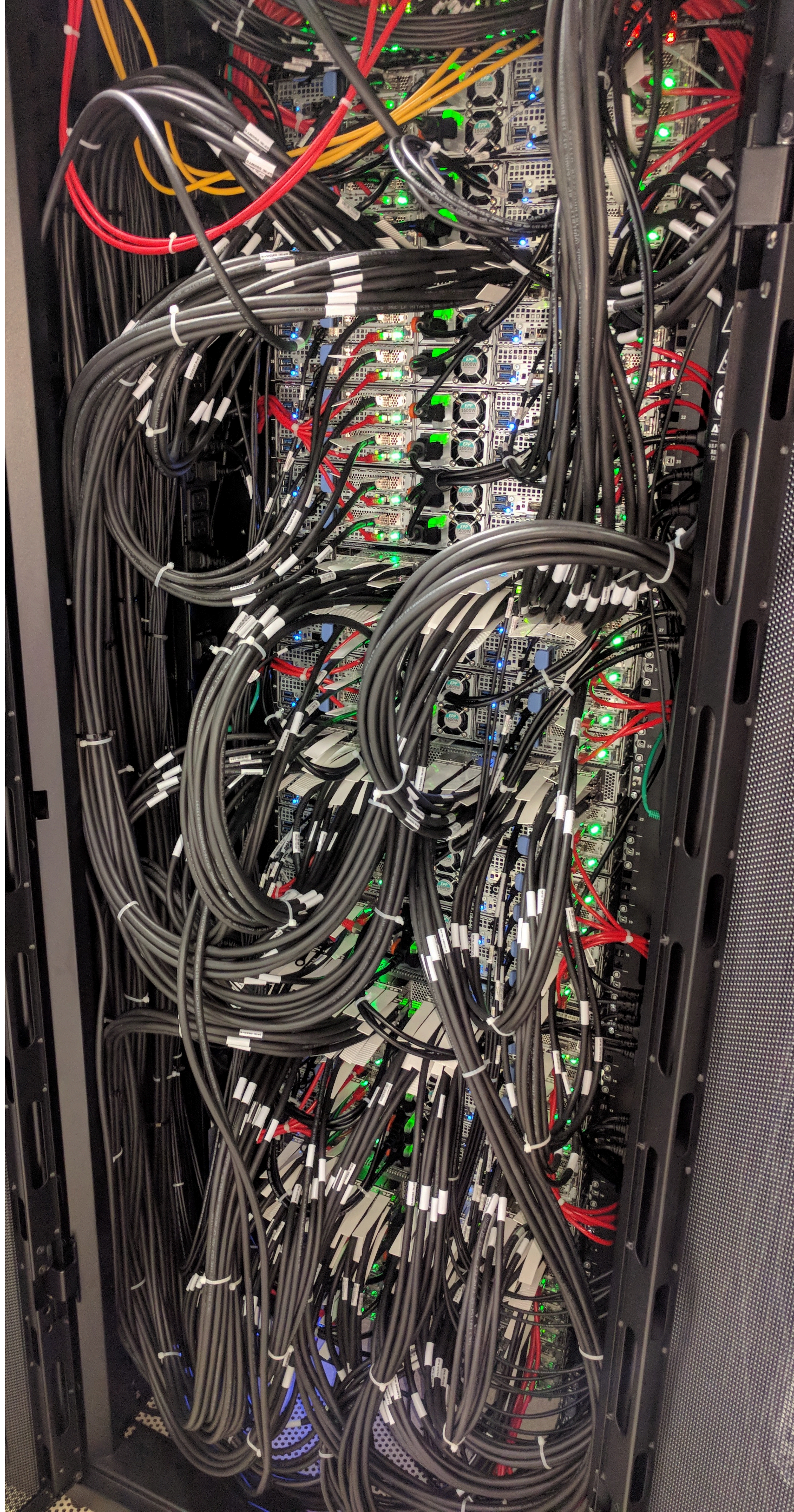
idea: test B-smoothness of the potential relations

What we needed for a record computation

- A fast sieving algorithm in dimension > 2.
- Identifying and removing duplicate relations.
- Adapting Schirokauer maps (virtual logarithms) to TNFS context.
- Glue-code to branch into CADO-NFS.
- A nice target: $\mathbb{F}_{p^6}$.

↑
in theory…

in practice… ⟶ grvingt



Our 521-bit record computation

$\mathbb{F}_{p_{87}^6}$

First boundary | Second boundary

0	Small char	$\frac{1}{3}$	Medium char	$\frac{2}{3}$	Large char	1	α
$\mathbb{F}_{2^{1024}}$		$\mathbb{F}_{p_5^{205}}$		$\mathbb{F}_{p_{30}^{34}}$		$\mathbb{F}_{p_{1024}}$	

Total computation time (core hours):

Relation Collection	Linear algebra	Schirokauer maps	Descent	Overall time
23,300	1,403	40	55	24,798

Focus on relation collection:

Parameters	[GGMT17]	[MR21]	This work
Algorithm	NFS	NFS	TNFS
Field size (bits)	422	423	521
Sieving dimension	3	3	6
Sieving time	201,600	69,120	**23,300**

[GGMT17]: L. Grémy, A. Guillevic, F. Morain, E. Thomé, Computing discrete logarithm in Fp6. Sac'17
[MR21]: G. McGuire, O. Robinson, Lattice Sieving in three dimensions for discrete log in medium characteristic. Journal of mathematical cryptology'21

A discrete logarithm

Finite field: $\mathbb{F}_{p^6}$ with 87-bit prime p, generator $g = x + \iota$

$$\text{target} = (31415926535897932384626433 + 83279502884197169399375105\iota + 82097494459230781640628620\iota^2) + x(89986280348253421170679821 + 48086513282306647093844609\iota + 55058223172535940812848111\iota^2)$$

$$\log(\text{target}) = 76272808168753222977667479701383785303538529763154 98$$

Thank you for your attention!

A discrete logarithm (in more details)

$p =$ 0x6fb96ccdf61c1ea3582e57 (87-bit prime) $\quad n = 6$

$$\mathbb{F}_{p^6} = \mathbb{F}_{p^3}[x]/(x^2 + 6441772330699146441962235 3x + 1)$$

Irreducible factor mod p, here f2

target = $a(\imath) + xb(\imath) \in \mathbb{F}_{p^6}$ with: $a(\imath), b(\imath)$ of degree 2 and coefficients $< p$.

$$\text{target} = (31415926535897932384626433 + 83279502884197169399375105\imath$$
$$+82097494459230781640628620\imath^2) + x(89986280348253421170679821$$
$$+48086513282306647093844609\imath + 55058223172535940812848111\imath^2)$$

generator = $x + \imath$

log(target) = 7627280816875322297766747970138378530353852976315498

Verification: $(x + \imath)^{\log(\text{target})} = \text{target} \pmod{\ell\text{-th powers}}$

Choice of subgroup

Initial target: $\mathbb{F}_{p^6}$ $\xrightarrow{\text{Pohlig-Hellman:}}$ Prime order subgroup of order $\ell \mid p^6 - 1$

We have the following factorisation: $p^6 - 1 = (p-1)(p+1)(p^2+p+1)(p^2-p+1)$

- $p - 1 = |\mathbb{F}_p^\times|$ If g and h are of order $\ell \mid p-1 \Rightarrow g, h \in \mathbb{F}_p^\times \Rightarrow$ NFS in $\mathbb{F}_p$ of 87 bits
- $p + 1 = |\mathbb{F}_{p^2}^\times| / |\mathbb{F}_p^\times|$ If g and h are of order $\ell \mid p+1 \Rightarrow g, h \in \mathbb{F}_{p^2}^\times \Rightarrow$ NFS in $\mathbb{F}_{p^2}$ of 175 bits
- $p^2 + p + 1 = |\mathbb{F}_{p^3}^\times| / |\mathbb{F}_p^\times|$ If g and h are of order $\ell \mid p^2+p+1 \Rightarrow g, h \in \mathbb{F}_{p^3}^\times \Rightarrow$ NFS in $\mathbb{F}_{p^3}$ of 261 bits
- $p^2 - p + 1$: 6th-cyclotomic subgroup Here, we can't go in a smaller subgroup…

Attention: it is not the largest subgroup!

Multiplicative group of a finite field

- The non-zero elements of a finite field form a multiplicative group.

- This group is cyclic, so all non-zero elements can be expressed as powers of a single element called a primitive element of the field.

Example 1: prime order finite fields: $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$

multiplicative group: $\mathbb{F}_p^\times = \{1, 2, \cdots, p-1\} = \mathbb{F}_p \backslash \{0\}$

Example 2: non-prime order finite fields: $\mathbb{F}_{p^n} \cong \mathbb{F}_p[X]/(P)$

--> elements are polynomials over $\mathbb{F}_p$ whose degree is less than n.

multiplicative group: $\mathbb{F}_{p^n}^\times = \{\text{invertible polynomials}\} = \mathbb{F}_{p^n} \backslash \{0\}$

Number field vs Function fields

Number field:

Finite extension of $\mathbb{Q}$

$\mathbb{Q} = \{p/q : p, q \text{ integers}\}$

$K = \mathbb{Q}[x]/(f)$

Example: $f = x^2 - d$

$K = \{x + y\sqrt{d} : x, y \in \mathbb{Q}\}$

Factor basis: prime ideals in $\mathcal{O}_K$

B-smoothness: compute norm of ideal = integer (from a resultant)

Function field:

Finite extension of $\mathbb{F}_p(\imath)$

$\mathbb{F}_p(\imath) = \{p(\imath)/q(\imath) : p(\imath), q(\imath) \in \mathbb{F}_p[\imath]\}$

$K = \mathbb{F}_p(\imath)[x]/(f)$

Example: $f = x^2 - (\imath^3 + 2\imath - 3)$

$K = \{x_0 + x_1\sqrt{\imath^3 + 2\imath - 3} : x_0, x_1 \in \mathbb{F}_p(\imath)\}$

Factor basis: prime ideals in $\mathcal{O}_K$

B-smoothness: compute norm of ideal = univariate polynomial (from a bivariate resultant)

Why do we choose a d-sphere?

Assumption: *size of norms depends only on size of vector coordinates.*

The norm for $c' \in C \backslash S_d(R)$ is greater than the norm for $c \in S_d(R)$.

When $d \to \infty$:

Difference in norms increases!

Conclusion: choosing $S_d(R)$ leads to smaller norms.