Calculating the Power Residue Symbol and Ibeta

Applications of Computing the Group Structure of the Principal Units of a $\mathfrak{p}$-adic Number Field Completion

Koen de Boer
Centrum Wiskunde & Informatica
Cryptology Group
Science Park 123
Amsterdam 1098 XG, The Netherlands

Carlo Pagano
Universiteit Leiden
Mathematisch Instituut
Niels Bohrweg 1
Leiden 2333CA, The Netherlands

ABSTRACT

In the recent PhD thesis of Bouw, an algorithm is examined that computes the group structure of the principal units of a $\mathfrak{p}$-adic number field completion. In the same thesis, this algorithm is used to compute Hilbert norm residue symbols. In the present paper, we will demonstrate two other applications.

The first application is the computation of an important invariant of number field completions, called ibeta. The algorithm that computes ibeta is deterministic and runs in polynomial time.

The second application uses Hilbert norm residue symbols and yields a probabilistic algorithm that computes the m-th power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ in arbitrary number fields K. This probabilistic algorithm relies on LLL-reduction and sampling of near-primes. Using heuristics, we analyse its complexity to be polynomial expected time in $n = [K : \mathbb{Q}]$, $\log|\Delta_K|$ and the input bit size – a tentative conclusion corroborated by timing experiments. An implementation of the algorithm in Magma will be available at https://github.com/kodebro/powerresiduesymbol.

1 INTRODUCTION

> *"Theorema fundamentale, quod sane inter elegantissima in hoc genere est referendum, in eadem forma simplici, in qua supra propositum est, a nemine hucusque fuit prolatum."*
>
> - C.F. Gauss, *Disquisitiones Arithmeticae*

The above quote is about the beautiful and famous quadratic reciprocity law. From this law, one can derive a classical algorithm that computes the quadratic residue symbol in $\mathbb{Z}$, also known as the Jacobi symbol.

The quadratic reciprocity law generalizes to higher powers, which is called the power residue reciprocity law [17, §VI.8]. Contrary to the quadratic case, no straightforward algorithm to compute higher power residue symbols follows from this law, mainly due to the fact that there occur Hilbert symbols in the reciprocity factor, for which there was – until very recently – no efficient algorithm known. Also, the lack of a Euclidean algorithm in most number fields fairly complicates the computation of the power residue symbol.

In Bouw's PhD thesis [5] an algorithm is proposed that computes Hilbert symbols effectively, which allows to compute higher power residue reciprocity – a significant improvement of ideas in Daberkow's article about Kummer extensions [8]. Using this effective reciprocity law, Squirrel gives an algorithm to compute the power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ for fixed m [23], relying on very heavy precomputations. The effective part of his algorithm – reducing the power residue symbol from arbitrary number fields to cyclotomic fields – is an idea proposed by Lenstra [15].

The master's thesis of one of the authors [2] introduces a probabilistic algorithm to compute the power residue symbol, that – under some heuristic assumptions – runs in expected time polynomial in the degree $n = [K : \mathbb{Q}]$, the logarithm of the absolute value of the field discriminant Δ_K and the size of the input. The algorithm is implemented in Magma [4] and seems to be practically feasible. This paper is partially a summary of the mentioned master's thesis.

Additionally, the algorithm in Bouw's thesis was the starting point of the research conducted by one of the authors on ibeta [20], a combinatorial invariant of local fields. This invariant turns out to parametrize the possible unit filtrations of local fields and connects the structure of the unit filtration with ramification theory and the possible jump sets of a character. These results, which we will briefly treat in this paper, indicate two main reasons why an implementation of an algorithm computing ibeta was desirable. The results led to further questions for which computer experimentation was eligible and, furthermore, the relations of ibeta with other invariants indicate that an implemented algorithm could be of practical use [20]. We describe in detail the algorithm and the key role of this algorithm in the theoretical questions that this subject has to offer.

The authors would like to thank J. Bouw for supplying an early version of his PhD thesis, prof. H.W. Lenstra for giving highly useful advise and dr. W. Bosma for his supervision.

2 PRELIMINARIES

Notation 2.1. In this paper, we write K for a number field, O_K for its ring of integers, $\mathfrak{a}$ for an ideal in O_K, $\mathfrak{p}$ for a prime ideal in O_K and n for the degree $[K : \mathbb{Q}]$. We will denote $\mathbb{F}_p$ for the unique finite field with p elements.

ISSAC '17, July 25-28, 2017, Kaiserslautern, Germany

978-1-4503-5064-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3087604.3087637

We assume that, during calculations, the ring of integers O_K is given by an integral basis: $O_K = \mathbb{Z}\gamma_1 + \ldots + \mathbb{Z}\gamma_n$. That means that every element $\alpha \in O_K$ can be uniquely represented by the coefficients of this basis:

$$\alpha = \sum_{i=1}^{n} c_i\gamma_i \text{ with } c_i \in \mathbb{Z}. \tag{1}$$

Despite of the fact that the power residue symbol algorithm also works in non-maximal orders $R \subseteq O_K$ [2], it is assumed – for simplicity – that the ring of integers is known.

Notation 2.2. We denote by $\mathbb{Q}_p$ the p-adic rationals, by $v_p : \mathbb{Q}_p \to \mathbb{Z}$ the p-valuation, by F a finite-degree extension of $\mathbb{Q}_p(\zeta_p)$ and by O the ring of integers of F with maximal ideal $\mathfrak{m}$. We denote the ramification index of F by $e = e(F/\mathbb{Q}_p)$ and the residue field degree by $f = f(F/\mathbb{Q}_p)$. We denote by $v : F \to \mathbb{Z}$ the valuation on F and by $\mathbb{F} = O/\mathfrak{m}$ the (finite) residue field of F. We choose an uniformizer $\pi \in O$, i.e. an element such that $(\pi) = \mathfrak{m}$. Also, we choose $\gamma \in \mathbb{F}$ such that every element in $\mathbb{F}$ can be uniquely written as $\sum_{i=0}^{f-1} c_i\gamma^i$ mod $\mathfrak{m}$, with $c_i \in \{0, \ldots, p-1\}$. We use the map $\bar{\cdot} : F \to \mathbb{F}$ for reducing modulo $\mathfrak{m}$.

We will use the notation $U_i(F) = 1 + \mathfrak{m}^i \subseteq O^*$ for the principal units of height i; these are elements of the form $1 + c \cdot \pi^i$, for an $i \in \mathbb{N}_{>0}$ and $c \in O$. Principal units $U_1(F)$ of height 1 are just called principal units. Denote by $\omega : \mathbb{F}^* \to F^*$ the Teichmüller map; for a definition see [9, Ex. I.13, p. 20].

3 PRINCIPAL UNITS OF A COMPLETION

Definition 3.1. An element $\delta \in U_{pe/(p-1)}(F)$ is called a weakly distinguished unit if it is not a p-th power in F.

According to [5, Prop. 7.21], one can compute a weakly distinguished unit efficiently in a given finite degree field extension F of $\mathbb{Q}_p(\zeta_p)$. Since weakly distinguished units $\delta \in F$ are not unique, it is assumed that – given an extension F – a fixed $\delta \in U_{pe/(p-1)}(F)$ is chosen beforehand.

Notation 3.2.
(i) We denote by J the index set $\{j \mid 1 \le j < pe/(p-1) \text{ and } p \nmid j\}$.
(ii) Let T_j be the set $\{1 - \omega(\gamma)^i\pi^j \mid 0 \le i < f\} \subseteq U_j$, for $j \in J$.
(iii) Let $T_{pe/(p-1)} = \{\delta\}$.
(iv) Let $T := \bigcup_{j \in J \cup \{pe/(p-1)\}} T_j$.

The following theorem is a short version of [5, §8.4, Th. 8.15].

Theorem 3.3. *Let F be a finite extension of $\mathbb{Q}_p(\zeta_p)$. Then the group homomorphism*

$$\phi : \mathbb{Z}_p^T \to U_1(F),\ (a_t)_{t\in T} \mapsto \prod_{t\in T} t^{a_t} \tag{2}$$

is a surjection, with kernel equal to $(\boldsymbol{b}) := (b_t)_{t\in T}\mathbb{Z}_p$ for some $\boldsymbol{b} = (b_t)_{t\in T} \in \mathbb{Z}_p^T$.

Remark 3.4. From the proof of this theorem (see [5, Th. 8.15] or [11, Th. 2.2]) one obtains that the isomorphism $U_1 \xrightarrow{\sim} \mathbb{Z}_p^T/(\mathbf{b})$ is effective. This means in particular that one can (within polynomial time with respect to the precision N, degree n and prime p) compute a p-adic approximation of $\mathbf{b} = (b_t)_{t\in T} \in \mathbb{Z}_p^T$ that satisfies

$$\prod_{t\in T} t^{b_t} \equiv 1 \bmod \mathfrak{m}^N \tag{3}$$

Remark 3.5. When one takes $N > \frac{pe}{p-1} + ke$ in Equation 3, the p-adic approximation of $b_t \in \mathbb{Z}_p$ has at least precision k for every $t \in T$ (see [5, §7.3] or [11, §2]). For the computation of ibeta it is enough to have precision $r + 1$ for all b_t, with r the maximum number such that $p^r \mid e$, the ramification index of $F : \mathbb{Q}_p$. This is a consequence of Theorem 4.4 and (I, β) being a extended jump set (as in Definition 4.2). So, in order to compute ibeta, one has to approximate $(b_t)_{t\in T}$ such that Equation 3 holds with precision $N > \frac{pe}{p-1} + (r+1)e$.

The following corollary is obtained from [5, Prop. 10.1].

Corollary 3.6. *For $\alpha, \beta \in F$, one can compute the Hilbert norm residue symbol $\left(\frac{\alpha,\beta}{\mathfrak{m}}\right)_m$ in time polynomially bounded by $n = [F : \mathbb{Q}_p]$ and $m \in \mathbb{N}$.*

4 COMPUTING IBETA

4.1 ibeta

In Theorem 3.3, a homomorphism $\mathbb{Z}_p^T \to U_1(F)$ is described, with kernel generated by an element $\mathbf{b} = (b_t)_{t\in T}$. Since this element depends heavily on the choices of $\pi \in \mathfrak{m}, \gamma \in \mathbb{F}$ and $\delta \in U_{pe/(p-1)}$, it is clearly not uniquely determined. In the current section, we examine how various invariants of the field extension $F : \mathbb{Q}_p(\zeta_p)$ are related with $\mathbf{b} \in \mathbb{Z}_p^T$.

As a $\mathbb{Z}_p$-module, the isomorphism type of $U_1(F)$ is completely determined by the degree $n = [F : \mathbb{Q}_p]$ and the number $k = v_p(\#\mu(F))$, where $\mu(F)$ are the roots of unity of F (see Theorem 3.3). This number k denotes the largest $k \in \mathbb{N}_{>0}$ such that the p^k-th root of unity ζ_{p^k} is contained in F. From Theorem 3.3 one can deduce that $k = \min_{t\in T} v_p(b_t)$.

By imposing more structure on $U_1(F)$, and seeing it as a *filtered* $\mathbb{Z}_p$-module with respect to the natural filtration $U_1(F) \supset U_2(F) \supset \ldots \supset U_i(F) \supset \ldots$, a much stronger relation emerges with the element $\mathbf{b}$ from Theorem 3.3. One will see shortly that the isomorphism type of $U_1(F)$ as a filtered $\mathbb{Z}_p$-module can be characterized by some 'reduced version' of $\mathbf{b}$, called ibeta.

The relation with the filtered module $U_1(F)$ and the element $\mathbf{b} \in \mathbb{Z}_p^T$ arises when one sees both $\mathbb{Z}_p^T$ and $U_1(F)$ as objects in the category of filtered $\mathbb{Z}_p$-modules $\mathbf{Mod}_{\mathrm{Filt}}(\mathbb{Z}_p)$. It is proved in [20] that there exists a 'natural filtration' on $\mathbb{Z}_p^T$ such that for any two surjective morphisms $\eta, \theta : \mathbb{Z}_p^T \to U_1$ in the category $\mathbf{Mod}_{\mathrm{Filt}}(\mathbb{Z}_p)$ holds that there exists an $\epsilon \in \mathrm{Aut}_{\mathrm{Filt}}(\mathbb{Z}_p^T)$ such that $\eta \circ \epsilon = \theta$. Here, $\mathrm{Aut}_{\mathrm{Filt}}(\mathbb{Z}_p^T)$ denotes the group of filtered automorphisms of $\mathbb{Z}_p^T$.

Therefore, the isomorphism type of $U_1(F)$ is encoded in the orbit of $\mathbf{b} \in \mathbb{Z}_p^T$ under the filtered automorphism group of $\mathbb{Z}_p^T$ with the mentioned natural filtration. It turns out that the sets of orbits of $\mathrm{Aut}_{\mathrm{Filt}}(\mathbb{Z}_p^T)$ acting on $\mathbb{Z}_p^T$ can be parametrized by *extended-$\rho_{(e,p)}$-jump-sets*.

Notation 4.1. Given e, p as in Notation 2.2. We denote by $\rho_{(e,p)} : \mathbb{Z} \to \mathbb{Z}$ the map $\rho_{(e,p)}(i) := \min(pi, i + e)$ and by $J^+ = J \cup \{\frac{pe}{p-1}\}$ with J as in Notation 3.2.

Definition 4.2. An extended ρ-jump-set is a pair (I, β) where $I \subset J^+$ and $\beta : I \to \mathbb{Z}_{\ge 1}$ such that β is strictly decreasing and the

map

$$i \mapsto \rho_{(e,p)}^{\beta(i)}(i) = \underbrace{\rho_{(e,p)} \circ \cdots \circ \rho_{(e,p)}}_{\beta(i) \text{ times}}(i)$$

is strictly increasing.

The jump set (I,β) corresponding to $\mathbf{b} \in \mathbb{Z}_p^T$ can be computed by applying Algorithm 2; here, $\mathbf{b}$ must be given with a sufficient precision as in Remark 3.5.

Definition 4.3. Let F be a finite extension of $\mathbb{Q}_p(\zeta_p)$. We denote by $(I_F,\beta_F) = \texttt{ibeta}(F)$ the jump set obtained by applying Algorithm 2 to $\mathbf{b} \in \mathbb{Z}_p^T$ from Theorem 3.3.

In [20] it is proved that the jump set (I_F,β_F) given by the output of Algorithm 2 is independent of the representation of the field F, making Definition 4.3 well-posed. Furthermore, this proof shows that (I_F,β_F) determines the orbit of $\mathbf{b} \in \mathbb{Z}_p^T$ under the filtered automorphism group of $\mathbb{Z}_p^T$ and therefore characterizes the isomorphism class of $U_1(F)$ as a filtered module. In other words, there is a one-to-one correspondence between isomorphism classes of filtered modules of the form $U_1(F)$ (with F a finite extension of $\mathbb{Q}_p(\zeta_p)$ having ramification index e) and *realizable* $\rho_{(e,p)}$-jump sets, as defined in Definition 4.5.

4.2 Properties of ibeta

In this section, three theorems about ibeta will be discussed. The proofs of these theorems can be found in [20].

Theorem 4.4. *Let p be a prime number, let $e \in \mathbb{Z}_{>0}$, and let (I,β) be an extended $\rho_{(e,p)}$-jump set as in Definition 4.2. Then the following are equivalent:*

- *There exists a finite extension $F : \mathbb{Q}_p(\zeta_p)$ with ramification index e, such that $(I,\beta) = (I_F,\beta_F) = \texttt{ibeta}(F)$;*
- *$p-1 \mid e$, $I \neq \emptyset$ and $\rho_{(e,p)}^{\beta(\min(I))}(\min(I)) = \frac{pe}{p-1}$.*

Definition 4.5. Let p be a prime number, let $e \in \mathbb{Z}_{>0}$. We call the $\rho_{(e,p)}$-jump set (I,β) realizable when the statements in Theorem 4.4 are true. Furthermore, denote

$$R_e = \{(I,\beta) \mid (I,\beta) \text{ is a realizable } \rho_{(e,p)}\text{-jump set}\}$$

Note that Theorem 4.4 states that the set R_e is effectively recognizable.

Notation 4.6. Let $e,f \in \mathbb{N}_{>0}$, and let $F = \mathbb{Q}_{p^f}(\zeta_p)$, where $\mathbb{Q}_{p^f}$ is the unique unramified extension of $\mathbb{Q}_p$ of degree f. As usual, $\mathfrak{m} = (1-\zeta_p)$ is the unique maximal ideal associated to F. We denote

$$E_{e,f} = \{g(x) \in F[x] \mid g(x) \text{ is } \mathfrak{m}\text{-Eisenstein, } \deg(g) = e\}.$$

Also, given an $\mathfrak{m}$-Eisenstein polynomial $g(x) \in F[x]$, we denote F_g for $F[x]/(g)$, the extension of F by adjoining a root of $g(x)$.

Theorem 4.7. *Let e,f and F as in Notation 4.6. Denote*

$$E_{(I,\beta)} = \{g \in E_{e,f} \mid \texttt{ibeta}(F_g) = (I,\beta)\}.$$

Then there exists an effectively computable probability function $G : R_e \to [0,1]$ from the set of realizable $\rho_{(e,p)}$-jump sets to the unit interval such that

$$\mu_H\left(E_{(I,\beta)}\right) = G((I,\beta)), \tag{4}$$

where μ_H is the Haar measure on $E_{e,f}$.

The measure on left hand side of Equation 4 is the Haar measure on the coefficients of the polynomials, which – by the Serre mass formula [22] – gives a natural counting measure on totally ramified relative extensions of fixed degree. So, informally, the value of the function $G(I,\beta)$ could be interpreted as the probability that a randomly chosen $g \in E_{e,f}$ (with respect to the Haar measure) satisfies $\texttt{ibeta}(F_g) = (I,\beta)$. The function $G(I,\beta)$ also has a natural combinatorial interpretation, which is interesting in itself [20].

Examining the proof of Theorem 4.7 carefully, one also obtains a non-probabilistic result: for certain special Eisenstein polynomials $g(x) \in F[x]$, $\texttt{ibeta}(F_g)$ can be determined by applying calculations on the coefficients of $g(x)$ as in Algorithm 1. Those special Eisenstein polynomials are called unsaturated.

Definition 4.8 (Unsaturated Eisenstein polynomials)**.** Given $F = \mathbb{Q}_{p^f}(\zeta_p)$ with unique maximal ideal $\mathfrak{m} = (1-\zeta_p)$ of $\mathcal{O} \subseteq F$. An $\mathfrak{m}$-Eisenstein polynomial $g(x) = x^d + \sum_{i=0}^{d-1} g_i x^i \in F[x]$ is called unsaturated iff there exists an $i \in \{1,\ldots,d-1\}$, coprime to p, such that $v_{\mathfrak{m}}(g_i) < p-1$.

Algorithm 1: Computes ibeta from an unsaturated Eisenstein polynomial

```
1  ibeta_special(g);
   Input  : An unsaturated m-Eisenstein polynomial g(x) of degree d.
   Output : Sequences I = (i_1, ..., i_k) and β = (β_1, ..., β_k) in Z^k
            related to the ramified extension F[x]/g(x).
2  Construct the set S = {(i, v_m(g_i)) | g_i ≠ 0} ;
3  Set the following lexicographic strict order ◁ on the set S:
   (i, m) ◁ (i', m') ⇔ (m < m' or (m = m' and i < i'))
4  Set k := 1 ;
5  while S ≠ ∅ do
6      s_k = (i_k, m_k) = min_◁ S ;            // minimum w.r.t. order ◁
7      Set S := {(i, m) ∈ S | v_p(i) < v_p(i_k)} ; // smaller p-valuation
8      k := k + 1 ;
9  end
10 I := { (d m_ℓ + i_ℓ) / p^{v_p(ℓ)} | ℓ ∈ {1, ..., k} } sorted increasing ;
11 β := {v_p(ℓ) + 1 | ℓ ∈ {1, ..., k}} sorted decreasing ;
```

Theorem 4.9. *Let $g(x) \in F[x]$ be an unsaturated $\mathfrak{m}$-Eisenstein polynomial as in Definition 4.8. Then Algorithm 1 computes the $\rho_{(e,p)}$-jump set $(I_{F_g},\beta_{F_g}) = \texttt{ibeta}(F_g)$ correctly.*

When $g \in F[x]$ is an unsaturated Eisenstein polynomial, one is able to compute from (I_{F_g},β_{F_g}) the so-called jumps of the ramification filtration of $F_g : F$, an important invariant of the field F_g, closely related to the Newton polygon of $g(\omega x + \omega)$ for ω a root of g [20]. This directly implies that this particular invariant is completely determined by the structure of the filtered module $U_1(F_g)$, for unsaturated Eisenstein extensions F_g.

4.3 Research on and applications of ibeta

4.3.1 Find a similar rule for saturated Eisenstein polynomials. For saturated Eisenstein polynomials, the proof of Theorem 4.7 is purely probabilistic, and nothing equivalent to Theorem 4.9 is known. This means that – for saturated Eisenstein polynomials – the only known

way to compute $\texttt{ibeta}(F_g)$ is via Algorithm 2. One might apply this algorithm to many saturated Eisenstein polynomials $g \in F[x]$ in order to find a structure in the computed $\texttt{ibeta}(F_g)$ for such $g \in F[x]$ or to discover a relation with important invariants from ramification theory.

4.3.2 Relate (I_F, β_F) to the Galois group Gal$(F/\mathbb{Q}_p(\zeta_p))$. Not only the relation with ramification theory is interesting; also finding a link between (I_F, β_F) and the Galois group $G_F = \mathrm{Gal}(F/\mathbb{Q}_p(\zeta_p))$ is a field of current research. The most interesting and challenging case is when $p \mid \#G_F$ and the extension $F : \mathbb{Q}_p(\zeta_p)$ is totally ramified. Algorithm 2 can be used to find such a link in these difficult cases.

The authors of this paper did already run Algorithm 2 in this context for some cases where G_F is isomorphic to the cyclic group of order p^k. These extension were of the form $F(\sqrt[p^k]{\pi}) : F = \mathbb{Q}_p(\zeta_p)$, with π a uniformizing element of F. Note that an Eisenstein polynomial defining this extension is always saturated.

4.3.3 Classifying jump sets of cyclic characters of Gal$(F/\mathbb{Q}_p(\zeta_p))$. In the theory of Galois representations, one might be interested in the jump sets of cyclic characters of $\mathrm{Gal}(\bar{F}/F)$, where $\bar{F}$ is the algebraic closure of the local field F. In [20] is proved that a list of all possible jump sets associated to such characters can be calculated explicitly from (I_F, β_F).

4.3.4 Relate $(I_{F'}, \beta_{F'})$ to (I_F, β_F) for an extension $F' : F : \mathbb{Q}_p(\zeta_p)$. There are already several theorems in this direction (see [20]), which are mostly results like in Theorem 4.10. The goal is to find a theorem that relates the change of (I_F, β_F) to $(I_{F'}, \beta_{F'})$ to an Eisenstein polynomial defining the ramified extension $F' : F$. With the implemented version of Algorithm 2 the authors are presently able to perform experiments in the direction of this goal. From these experiments, one might expect to find structures that imply a 'relative' version of Algorithm 1.

Theorem 4.10. *Suppose $F' : F : \mathbb{Q}_p(\zeta_p)$ are finite extensions such that $F' : F$ is totally ramified and of degree p. Let $i < j$ be consecutive indices in I_F such that $\beta_F(i) - \beta_F(j) \geq 2$.*

Then $i, j \in I_{F'}$, $\beta_{F'}(i) = \beta_F(i) + 1$ and $\beta_{F'}(j) = \beta_F(j) + 1$.

Algorithm 2: Computes ibeta from $\mathbf{b} \in \mathbb{Z}_p^T$

```
ibeta(b);
Input  : An element b = (b_t)_{t∈T} ∈ Z_p^T with for every b_t precision at
         least p^{r+1}, with r = v_p(e).
Output : I = (i_1, ..., i_k) and β = (β_1, ..., β_k) in Z^k.
// Initialization
Set I := {i | 1 ≤ i < pe/(p−1) and p ∤ i} ∪ {pe/(p−1)} ;
Set T_i := {1 − ω(γ)^j π^i | 0 ≤ j < f} ⊆ U_i, for i ∈ I\{pe/(p−1)} ;
Set T_{pe/(p−1)} = {δ} ;
for i ∈ I do
    Compute m = min{v_p(b_t) | t ∈ T_i} ;
    if m > r + 1 then
        Remove i from I ;
    else
        Set β_i = m ;
    end
end
// 'Upwards reduction'
for i ∈ I do
    I := I\{i' ∈ I | i < i' and β_i ≤ β_{i'}} ;
end
// 'Downwards reduction'
for i' ∈ I do
    I := I\{i ∈ I | i < i' and i · p^{β_i} ≥ i' · p^{β_{i'}}} ;
end
Return I, (β_i | i ∈ I) ;
```

5 COMPUTING THE POWER RESIDUE SYMBOL

5.1 Preliminaries

Notation 5.1. In this section, K is a number field containing a primitive m-th root of unity $\zeta_m \in K$, and $\mathfrak{p}$ is a prime ideal of O_K, coprime to m. Also, we denote $\mu_m = \langle \zeta_m \rangle$, for the group of m-th roots of unity in K.

The following definitions of the power residue symbol can be found in [12, p. 111].

Definition 5.2 (Power residue symbols above prime ideals)**.** Let $\alpha \in O_K \backslash \mathfrak{p}$. Then we define $\left(\frac{\alpha}{\mathfrak{p}}\right)_m \in \mu_m$ to be the unique m-th root of unity that satisfies

$$\left(\frac{\alpha}{\mathfrak{p}}\right)_m \equiv \alpha^{\frac{N(\mathfrak{p})-1}{m}} \bmod \mathfrak{p}. \tag{5}$$

General power residue symbols – i.e., above any ideal – are just multiplicative continuations of Definition 5.2. Since the ring of integers of a number field is a Dedekind ring, every (fractional) ideal can be decomposed uniquely into a product of prime ideals:

$$\mathfrak{b} = \prod_{\mathfrak{p} \mid \mathfrak{b}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{b})}.$$

Definition 5.3 (Power residue symbol)**.** Let K be as in Notation 5.1, let $\mathfrak{b}$ an ideal of O_K coprime to m and let $\alpha \in O_K$ be an element coprime to $\mathfrak{b}$. We define

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_m := \prod_{\mathfrak{p} \mid \mathfrak{b}} \left(\frac{\alpha}{\mathfrak{p}}\right)_m^{v_{\mathfrak{p}}(\mathfrak{b})}.$$

Lemma 5.4. *The power residue symbol has the following properties, for all $\alpha, \beta, \gamma \in K$, for all ideals $\mathfrak{b}, \mathfrak{c}$ of O_K and for all prime ideals $\mathfrak{p}$ of O_K, provided that the numerator and de denominator in the symbol are coprime, and the denominator is coprime to m.*

$$\left(\frac{\alpha}{\beta}\right)_m \left(\frac{\beta}{\alpha}\right)_m^{-1} = \prod_{\mathfrak{p} \mid m\infty} \left(\frac{\alpha, \beta}{\mathfrak{p}}\right)_m \text{ (reciprocity law)}^1 \tag{6}$$

$$\left(\frac{\alpha}{\mathfrak{b}}\right)_m = \left(\frac{\alpha + \beta}{\mathfrak{b}}\right)_m \text{ for every } \beta \in \mathfrak{b}. \tag{7}$$

$$\left(\frac{\alpha\beta}{\mathfrak{b}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \cdot \left(\frac{\beta}{\mathfrak{b}}\right)_m \text{ and } \left(\frac{\alpha}{\mathfrak{bc}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \cdot \left(\frac{\alpha}{\mathfrak{c}}\right)_m \tag{8}$$

[1] The infinity sign in this equation is about so-called infinite primes. Only the Hilbert symbols above the 'real' infinite primes $\sigma : K \to \mathbb{R}$ are non-trivial and can easily be computed by observing the signs of $\sigma(\alpha)$ and $\sigma(\beta)$ [17, § III.5].

$$\left(\frac{\alpha\gamma^m}{\mathfrak{b}}\right)_m = \left(\frac{\alpha}{\mathfrak{b}}\right)_m \tag{9}$$

Proof. A proof of these properties can be found in [17, §VI.8.3] and [10, Ch. 2]. □

Notation 5.5. In this paper, the right hand side of Equation 6 will be denoted by $U(\alpha,\beta)$, the *Umkehrfaktor* (German for inversion factor):

$$U(\alpha,\beta) = \prod_{\mathfrak{p}|m\infty} \left(\frac{\alpha,\beta}{\mathfrak{p}}\right)_m.$$

Here, the symbol $\left(\frac{\alpha,\beta}{\mathfrak{p}}\right)_m$ denotes the m-th Hilbert norm residue symbol at the prime $\mathfrak{p}$.

Remark 5.6. An immediate corollary of the results of [5] is that the Umkehrfaktor can be computed in polynomial time (see Corollary 3.6), implying that the reciprocity law (see Equation 6) can be used extensively in an algorithm that computes the principal power residue symbol.

So, from now on, we assume that one can compute $\left(\frac{\beta}{\alpha}\right)_m$ from $\left(\frac{\alpha}{\beta}\right)_m$ in polynomial time, with the following calculation.

$$\left(\frac{\beta}{\alpha}\right)_m = U(\alpha,\beta)^{-1}\left(\frac{\alpha}{\beta}\right)_m.$$

5.2 Computation of the power residue symbol

The straightforward way to compute $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ is by factoring $\mathfrak{b}$ into prime ideals and using formula (5) to evaluate the symbol $\left(\frac{\alpha}{\mathfrak{p}}\right)_m$ above each prime ideal dividing $\mathfrak{b}$. Since factoring ideals is as hard as factoring integers, for which the fastest known algorithm is still only subexponential [21], [13], this is not the right approach.

Instead, we can use the techniques of [2], in which the power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ is computed using three stages:

- Principalization, which is essentially reducing the computation of $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ to $\left(\frac{\alpha}{\beta}\right)_m$ for some $\beta \in \mathfrak{b}$.
- Reduction, an optional stage, which reduces the computation of $\left(\frac{\alpha}{\beta}\right)_m$ for large $\alpha,\beta \in K$ to many 'smaller' $\left(\frac{\alpha_i}{\beta_i}\right)_m$ with $\alpha_i,\beta_i \in K$.
- Evaluation, where $\left(\frac{\alpha'}{\beta'}\right)_m$ is computed directly, using a trick that relies on prime density results.

5.2.1 Principalization and evaluation. Since the stages principalization and evaluation are very alike, we will treat both of them in one subsection. Theoretically, these two stages are the most important, whereas practically, the reduction stage reduces the running time drastically.

In these two stages, the notion of B-near prime ideals is used. A near prime ideal has a norm that is the product of one single large prime and several other very small primes. More formally:

Definition 5.7 (B-near prime number)**.** An integer $N \in \mathbb{N}$ is said to be a B-near prime number if N factorizes as follows:

$$N = p \cdot \prod_{i=1}^{k} p_i \text{ with } p_i \le B \text{ for all } 1 \le i \le k,$$

where $\{p_i\}$ may contain repeating primes.

Definition 5.8 (B-near prime ideal)**.** An ideal $\mathfrak{a}$ of R is called a B-near prime ideal when the norm $N(\mathfrak{a})$ is a B-near prime number as in Definition 5.7.

Remark 5.9. The computational advantage of B-near prime ideals is that they are both effectively recognizable and factorizable when B is polynomially bounded by the degree $n = [K:\mathbb{Q}]$, as explained below.

For B polynomially bounded in the degree $n = [K:\mathbb{Q}]$, checking whether an ideal $\mathfrak{a}$ is B-near prime can be done by a polynomial time algorithm. Compute the norm $N = N(\mathfrak{a})$, and apply trial division up to B to the number N, i.e. $N = r \cdot \prod_{i=1}^{k} p_i$ with $p_i \le B$. Then, use a fast primality proving algorithm to check whether r is prime or not. The ideal $\mathfrak{a}$ is a B-near prime if and only if r is prime. Since primality proving can be done in polynomial time [1], above procedure recognizes B-near prime ideals in polynomial time.

Also, if B is of polynomial size in the degree $n = [K:\mathbb{Q}]$, the B-near prime ideals $\mathfrak{a}$ are effectively factorizable, since one can find the prime factorization of the norm. Write $N(\mathfrak{a}) = p \cdot \prod_{i=1}^{k} p_i^{m_i}$ with $p_i \le B$ and $p_i \ne p_j \ne p$ for $i \ne j$. Here, all p, p_i are prime.

(a) Set $\mathfrak{p}_p := (\alpha, p)$.
(b) Factor $(\alpha, p_i^{m_i}) = \prod_{j=1}^{k_i} \mathfrak{p}_{p_i,j}^{t_j}$;
(c) Then the prime ideal factorization of (α) is:

$$(\alpha) = \mathfrak{p}_p \prod_{i=1}^{k} \prod_{j=1}^{k_i} \mathfrak{p}_{p_i,j}^{t_j}. \tag{10}$$

Principalization.
The principalization algorithm (Algorithm 3), consists of sampling 'random', relatively small elements $\beta \in \mathfrak{b}$, and hoping that the ideal $\mathfrak{c} = (\beta)/\mathfrak{b}$ is a B-near prime ideal. Such B-near prime ideals are easily factorizable, and one calculates $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$ by computing $\left(\frac{\alpha}{\beta}\right)_m \cdot \left(\frac{\alpha}{\mathfrak{c}}\right)_m^{-1}$, where in the computation of $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$, the factorization of $\mathfrak{c}$ is used.

See Algorithm 3 for a description of this algorithm.

Evaluation.
The idea behind Algorithm 4 is to repeatedly multiply α_0 by the m-th power of random γ, until $\gamma^m\alpha_0 \bmod \beta$ has a small representative $\hat{\alpha} \in R$ that generates a B-near prime ideal (as in Definition 5.8, with a polynomially bounded B).

After finding such an element $\hat{\alpha}$ generating a near-prime ideal, one can use reciprocity, which reduces the computation of $\left(\frac{\hat{\alpha}}{\beta}\right)_m = \left(\frac{\alpha}{\beta}\right)_m$ to the calculation of $\left(\frac{\beta}{\hat{\alpha}}\right)_m$. Since one can effectively (probabilistically) factorize B-near prime ideals, one obtains $(\hat{\alpha}) = \mathfrak{p}_p \cdot \prod_{i=1}^{k} \mathfrak{p}_i$ and computes

$$\left(\frac{\beta}{\hat{\alpha}}\right)_m = \left(\frac{\beta}{\mathfrak{p}_p}\right)_m \cdot \prod_{i=1}^{k} \left(\frac{\beta}{\mathfrak{p}_i}\right)_m.$$

Algorithm 3: Principalization: reducing the general power residue symbol to the principal power residue symbol

1 PowerResidueSymbol($\alpha, \mathfrak{b}$);
Input : An element α in R and an ideal $\mathfrak{b}$ in O_K
Output: The power residue symbol $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$
2 Set $n := [K : \mathbb{Q}]$, where K is the quotient field of O_K ;
3 Set $B := 12\log^2(|\Delta_K|)$ for the bound for near-primeness (see Remark 5.10);
4 Compute an LLL-reduced basis $(\beta_1, \ldots, \beta_n)$ of $\mathfrak{b}$;
5 **repeat**
// Pick a random but small element from $\mathfrak{b}$
6 Pick a random vector $(c_1, \ldots, c_n) \in \mathbb{Z}^n$, with $|c_i| \leq 3$ for all i ;
7 Set $\beta := \sum_{i=1}^n c_i\beta_i$;
8 Calculate $N := N(\beta)/N(\mathfrak{b})$;
9 **until** N *is a* B*-near prime number*;
// N is of the form $p \cdot \prod_{i=1}^{r'} p_i$ for 'small' p_i now
10 Calculate the ideal $\mathfrak{c} := (\beta)/\mathfrak{b}$, using (for example) [6, §4.8.4] ;
11 Factorize $\mathfrak{c} := \mathfrak{p} \cdot \prod_{i=1}^r \mathfrak{p}_i$, using the factorization of N as in Equation 10;
12 Compute $\left(\frac{\alpha}{\mathfrak{c}}\right)_m$ using above factorization ;
13 Compute $\left(\frac{\alpha}{\beta}\right)_m$ with reduction (subsubsection 5.2.2) and evaluation (Algorithm 4) ;
14 Return $\left(\frac{\alpha}{\beta}\right)_m \left(\frac{\alpha}{\mathfrak{c}}\right)_m^{-1}$;

Remark 5.10. In line 11 of Algorithm 4 and line 3 of Algorithm 3, the bound $B = 12\log^2(|\Delta_K|)$ is used, where Δ_K is the discriminant of K. This particular choice – which may be increased to some other fixed bound polynomial in $n = [K : \mathbb{Q}]$ and $\log(|\Delta_K|)$ – is based on practical experiments. One might heuristically assume that B satisfies Assumption 5.18.

5.2.2 Reduction. The reduction stage is optional and is only useful when (A) the input variables α, β are both large, or (B) one of α, β is much larger than the other.

In case (B), the solution is easy. Assume α is much larger than β. Then compute an LLL-reduced basis M_β of the ideal (β), and reduce α modulo M_β as in [7, Algorithm 1.4.13, p. 33]. One hopes that, after this procedure, α and β are about the same size.

In case (A), the inputs are both (very) large, say, with coefficients with bit size $g(n) > 6n$, where $n = [K : \mathbb{Q}] \geq 2$. Again, assume α to be larger than β. Now, one can examine the lattice L:

$$L = \{(\gamma_1, \gamma_2) \in O_K \times O_K \mid \gamma_1\alpha - \gamma_2 \in (\beta)\}.$$

One can deduce that this lattice has discriminant $N(\beta)$, and dimension $2n$. Applying LLL-reduction, one finds a short vector $(\gamma_1, \gamma_2) \in O_K^2$ with:

$$\sqrt{\|\gamma_1\|^2 + \|\gamma_2\|^2} \leq 2^n N(\beta)^{\frac{1}{2n}} \approx 2^n \|\beta\|^{1/2}$$
$$\approx 2^{\frac{g(n)+2n}{2}} \leq 2^{\frac{2g(n)}{3}} \approx \|\beta\|^{\frac{2}{3}},$$

where heuristically is assumed that $\|\beta\| \approx N(\beta)^{\frac{1}{n}}$, and $\|\beta\| \approx 2^{g(n)}$.

Using the results of Lemma 5.4, one can deduce that $\left(\frac{\alpha}{\beta}\right)_m = \left(\frac{\gamma_2}{\beta}\right)_m \left(\frac{\gamma_1}{\beta}\right)_m^{-1}$. So, the computation of the original power residue symbol is reduced to the computation of two power residue symbols with smaller input. By using reciprocity (Equation 6) and using the fact that γ_i is smaller then β, one can reduce the symbols $\left(\frac{\gamma_i}{\beta}\right)_m$ for $i = 1, 2$ even further. This results in a so-called reduction tree.

The exact details and various improvements of above algorithm can be found in [2, Ch. 4].

Algorithm 4: Evaluation; computing the principal power residue symbol

1 PrincipalPowResSym(α_0, β);
Input : Elements $\alpha_0, \beta \in O_K$.
Output: The power residue symbol $\left(\frac{\alpha_0}{\beta}\right)_m$
2 Set n as the degree of the number field of O_K ;
3 **repeat**
4 **repeat**
5 Take a random $\bar{\gamma} \in O_K/\beta$;
6 Set $\alpha := \alpha_0 \cdot \bar{\gamma}^m$ modulo β, with modular exponentiation ;
7 **until** α *is invertible modulo* β;
8 Find $\bar{\alpha}$, a small representative of α modulo β, as as in [7, Algorithm 1.4.13, p. 33] ;
9 Lift $\bar{\alpha}$ coordinate-wise to O_K, call it $\hat{\alpha}$;
10 Calculate its norm, $N := N(\hat{\alpha})$;
11 Factorize $N = \left(\prod_{i=1}^k p_i\right) \cdot r$ using trial division with bound $B = 12\log^2(|\Delta_K|)$;
// I.e. $p_i \leq B$ for all i, and p_i are primes
12 **until** r *is prime and* N *is coprime with* m;
// r is prime, and $\hat{\alpha}$ is invertible mod β
13 Set $\mathfrak{p}_r = (\alpha, r)$;
14 Factorize the ideal $(\alpha) = \mathfrak{p}_r \cdot \prod_{i=1}^s \mathfrak{p}_i$, using the factorization of N, as in Equation 10 ;
15 Calculate the Umkehrsymbol $U(\hat{\alpha}, \beta)$;
16 Return $\prod_{i=1}^s \left(\frac{\beta}{\mathfrak{p}_i}\right)_m \cdot \left(\frac{\beta}{\mathfrak{p}_r}\right)_m \cdot U(\hat{\alpha}, \beta)$;

5.3 Analysis

We focus on the analysis of the principalization and the evaluation stage.

5.3.1 Size of reduced elements. In both Algorithm 3 (line 6) and Algorithm 4 (line 8) 'small' elements of the form $\sum_{i=1}^n c_i\beta_i$ occur. In this section a bound on such elements is derived, using properties of LLL-reduced bases [14]. Consider the embedding

$$K \xrightarrow{\Psi} K_{\mathbb{R}} \subseteq \prod_{\sigma:K\to\mathbb{C}} \mathbb{C}, x \mapsto (\sigma(x))_\sigma$$

where $\sigma : K \to \mathbb{C}$ ranges over the embeddings of K in $\mathbb{C}$. Seeing $\Psi(O_K)$ and $\Psi(\mathfrak{a})$ as lattices in $K_{\mathbb{R}}$ (for ideals $\mathfrak{b}$ of O_K), and using the standard Hermitian inner product on $\prod_{\sigma:K\to\mathbb{C}} \mathbb{C} \supseteq K_{\mathbb{R}}$ (see [24, p. 52]), one obtains

$$\Delta(\Psi(O_K)) = \sqrt{|\Delta_K|} \text{ and } \Delta(\Psi(\mathfrak{b})) = N(\mathfrak{a})\sqrt{|\Delta_K|}.$$

Also, using the arithmetic/geometric mean inequality, we have

$$N(\alpha)^{\frac{1}{n}} = \left(\prod_{\sigma:K\to\mathbb{C}} \sigma(\alpha)\right)^{\frac{1}{n}} \leq \frac{1}{n} \sum_{\sigma:K\to\mathbb{C}} |\sigma(\alpha)|$$

$$\leq \frac{1}{\sqrt{n}} \sqrt{\sum_{\sigma: K \to \mathbb{C}} |\sigma(\alpha)|^2} = \frac{1}{\sqrt{n}} \|\alpha\| \tag{11}$$

Applying LLL-reduction in the lattice $\Psi(\mathfrak{b})$ results [18, Ch. 2] in a basis $(\beta_1, \ldots, \beta_n)$ of $\mathfrak{b}$ satisfying the following bound:

$$\prod_{i=1}^{n} \|\beta_i\| \leq 2^{\frac{n(n-1)}{2}} |\Delta_K|^{\frac{1}{2}} N(\mathfrak{b}). \tag{12}$$

Using Equation 11 and Equation 12, this yields

$$\left(\prod_{i=1}^{n} N(\beta_i) \right)^{\frac{1}{n}} \leq \left(\prod_{i=1}^{n} \frac{\|\beta_i\|^n}{n^{\frac{n}{2}}} \right)^{1/n} \leq n^{-n/2} \cdot 2^{\frac{n(n-1)}{2}} \cdot |\Delta_K|^{\frac{1}{2}} \cdot N(\mathfrak{b}).$$

Proposition 5.11. *Let $(\beta_1, \ldots, \beta_n)$ be an LLL-reduced basis of $\mathfrak{b}$, an ideal of O_K. For elements of the form $\beta = \sum_{i=1}^{n} c_i \beta_i$ with $|c_i| \leq C$ and $c_i \in \mathbb{Q}$, we have the following bound:*

$$\log N(\beta) \leq n \left(\log C + \log n + 2n(n-1) + \frac{\log |\Delta_K|}{2} + \log N(\mathfrak{b}) \right).$$

Proof. A proof can be found in the appendix of the full version of this paper [3]. □

Remark 5.12. The above bound is really pessimistic. In most cases, we have

$$N(\beta) \leq 2^{\frac{n(n-1)}{2}} \cdot |\Delta_K|^{\frac{1}{2}} \cdot N(\mathfrak{b}).$$

Remark 5.13. Note that the coefficients c_i might be rational, as is the case in Algorithm 4. Also, with the Proposition 5.11, one is able to generate many 'relatively small' elements in $\mathfrak{b}$, as is needed in Algorithm 3.

5.3.2 Density of B-near primes. As we will see later on, the expected running time of the principalization and evaluation algorithm depends on the density of B-near primes in a fixed class of the ideal class group. The following result follows if one applies [16, §7.2, Prop. 7.17, p. 347] to the set

$$A := \{\mathfrak{p} \text{ prime ideal of } O_K \mid \mathfrak{p} \text{ is in the ideal class } [\mathfrak{g}]\},$$

where $[\mathfrak{g}] \in Cl(K)$ is an arbitrary ideal class. The set A is a regular set of prime ideals (see for example [17, §13, Thm. 13.2]) with Dirichlet density $\frac{1}{h_K}$, where $h_K = \#Cl(K)$ is the class number of K.

Theorem 5.14. *If K is a number field, then*

$$\sum_{\substack{\mathfrak{p} \in [\mathfrak{g}] \\ N(\mathfrak{p}) \leq x}} 1 \sim \frac{x}{h_K \log x} \text{ as } x \to \infty, \tag{13}$$

where $\mathfrak{p}$ ranges over all prime ideals of O_K in the ideal class $[\mathfrak{g}]$.

Notation 5.15 (Class number formula)**.** Denote

$$\rho_K := \frac{2^{r_1 + r_2} \pi^{r_2} R_K h_K}{w_K \sqrt{|\Delta_K|}},$$

where r_1 is the number of real embeddings of K, r_2 is the number of pairs of complex embeddings of K, R_K is the regulator of K (see for example [17, p. 42-43]), w_K is the number of roots of unity in K, Δ_K is the discriminant of O_K and $h_K = \#Cl(K)$, the class number of K. The number ρ_K equals the residue at $s = 1$ of the Dedekind zeta function $\zeta_K(s)$ of the number field K.

The following theorem counts the number of ideals in a particular ideal class $[\mathfrak{g}]$ that have norm bounded by x, and is obtained from [19, §9.5, Prop. 9.17].

Theorem 5.16. *For K a number field, and $[\mathfrak{g}]$ an ideal class in $Cl(K)$, we have*

$$\sum_{\substack{N(\mathfrak{a}) \leq x \\ \mathfrak{a} \in [\mathfrak{g}]}} 1 \sim \frac{\rho_K}{h_K} x \text{ as } x \to \infty$$

Notation 5.17. For $B \in \mathbb{N}$ we define the B-smooth ideals of O_K by

$$\mathcal{I}(B) := \{\mathfrak{a} \text{ ideal of } O_K \mid N(\mathfrak{a}) \text{ is a } B\text{-smooth number }\},$$

where a B-smooth number is a number whose prime factorization only contains primes numbers $\leq B$.

To prove the next lemma, we need the following heuristic assumption.

Assumption 5.18. *There exists a $B_K \in \mathbb{N}$ bounded polynomially in $\log |\Delta_K|$ and $n = \deg K$ such that, for all number fields K, one has*

$$\sum_{N(\mathfrak{a}) \leq B_K} \frac{1}{N(\mathfrak{a})} \geq \rho_K.$$

Lemma 5.19. *Suppose K is a number field, $[\mathfrak{g}] \in Cl(K)$ is an ideal class and $B = B_K$ as in Assumption 5.18. Then*

$$\#\{\mathfrak{a} \in [\mathfrak{g}] \text{ ideal of } O_K \mid \mathfrak{a} \text{ is a } B\text{-near prime}, N(\mathfrak{a}) \leq x\} \gtrsim \frac{\rho_K x}{h_K \log x}. \tag{14}$$

Proof. We have

$$\begin{aligned} &\#\{\mathfrak{a} \in [\mathfrak{g}] \text{ ideal of } O_K \mid \mathfrak{a} \text{ is a B-near prime}, N(\mathfrak{a}) \leq x\} \\ &= \sum_{\substack{\mathfrak{a} \in \mathcal{I}(B) \\ N(\mathfrak{a}) \leq x}} \sum_{\substack{\mathfrak{p} \in [\mathfrak{g}][\mathfrak{a}]^{-1} \\ N(\mathfrak{p}) \leq x/N(\mathfrak{a})}} 1 \sim \sum_{\substack{\mathfrak{a} \in \mathcal{I}(B) \\ N(\mathfrak{a}) \leq x}} \frac{x/N(\mathfrak{a})}{h_K(\log x - \log N(\mathfrak{a}))} \\ &\geq \frac{x}{h_K \log x} \sum_{\substack{\mathfrak{a} \in \mathcal{I}(B) \\ N(\mathfrak{a}) \leq x}} \frac{1}{N(\mathfrak{a})} \geq \frac{\rho_K x}{h_K \log x}. \end{aligned} \tag{15}$$

□

Corollary 5.20. *Suppose K is a number field, $[\mathfrak{g}] \in Cl(K)$ is an ideal class and B_K as in Assumption 5.18. Then, asymptotically, the fraction of B_K-near primes in the set $\{\mathfrak{a} \in [\mathfrak{g}] \mid N(\mathfrak{a}) \leq x\}$ is at least $\frac{1}{\log x}$.*

Proof. See the appendix of the full version of this paper [3]. □

5.3.3 Evaluation analysis. The 'loop' part of Algorithm 4, i.e. lines 3–12, is the most difficult part to analyse, since it is not clear when this loop terminates. The main question is: how often is $N(\hat{\alpha})$ a B-near prime number (as in Definition 5.7)?

Assumption 5.21. *Let*

$$M = \exp\left(n \left(\log C + \log n + 2n(n-1) + \frac{\log |\Delta_K|}{2} + \log N(\mathfrak{b}) \right) \right),$$

where the right side is obtained from Proposition 5.11. Then the sampling of $(\hat{\alpha})$ from lines 3–12 of Algorithm 4 happens uniformly distributed in the set

$$S = \{\mathfrak{a} \text{ ideal of } O_K \mid \mathfrak{a} \text{ principal and } N(\mathfrak{a}) \leq M\}$$

With above assumption and using Corollary 5.20, the probability that Algorithm 4 finds a B-near prime approximately equals

$$\mathbb{P}[(\hat{\alpha}) \text{ is a principal } B\text{-near prime ideal with norm bounded by } M] = \frac{1}{\log M} = \frac{1}{n\left(\log C + \log n + 2n(n-1) + \frac{\log|\Delta_K|}{2} + \log N(\mathfrak{b})\right)}$$

This means that one expects to execute the loop part of Algorithm 4 (lines 3–12) around $\log M$ times, a number polynomially bounded by n, $\log|\Delta_K|$ and $\log N(\mathfrak{b})$. So, under the mentioned assumptions, one might suggest that Algorithm 4 is a polynomial expected time algorithm.

5.3.4 Principalization analysis. The analysis of the principalization algorithm is very similar to that of the evaluation algorithm. Suppose one wants to apply the principalization algorithm on $\left(\frac{\alpha}{\mathfrak{b}}\right)_m$. Instead of searching *principal* B-near prime ideals, one tries to find B-near prime ideals $\mathfrak{c}$ in the ideal class $[\mathfrak{b}]^{-1} \in \mathrm{Cl}(K)$. This is accomplished by finding 'relatively small' $\beta \in \mathfrak{b}$ and define $\mathfrak{c} = (\beta)/\mathfrak{b}$.

Assumption 5.22. *Let*

$$M = \exp\left(n\left(\log C + \log n + 2n(n-1) + \frac{\log|\Delta_K|}{2} + \log N(\mathfrak{b})\right)\right),$$

where the right side is obtained from Proposition 5.11. Then the sampling of $\mathfrak{c}$ *from lines 5–9 of Algorithm 3 happens uniformly distributed in the set*

$$S = \{\mathfrak{a} \text{ ideal of } O_K \mid \mathfrak{a} \in [\mathfrak{b}]^{-1} \text{ and } N(\mathfrak{a}) \leq M\}$$

As in the previous reasoning about the evaluation algorithm, one expects to find a B-near prime ideal after executing the loop in lines 5–9 of Algorithm 3 approximately $\log M$ times. This, again, might suggest that the principalization part of the algorithm has expected polynomial running time. With that, one might imply that the overall running time of the power residue symbol algorithm proposed in this paper has expected polynomial running time.

6 RESULTS

In order to strengthen one's believe in the above heuristic analysis, we provide in Figure 1 some timings of our implementation of the power residue symbol algorithm in Magma, applied to various cyclotomic fields.

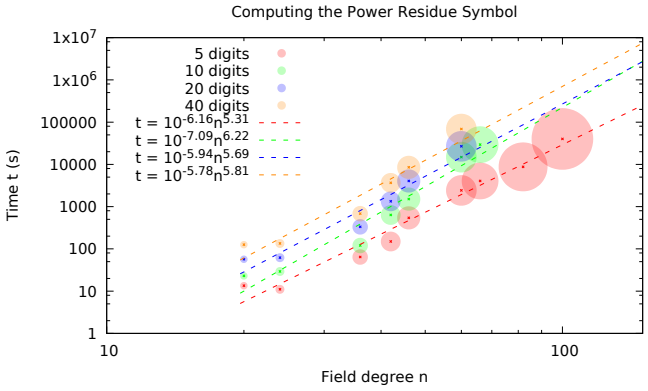

Figure 1: Timings of an implementation of the power residue symbol algorithm in Magma, for the fields $K = \mathbb{Q}(\zeta_m)$ for $m = 25, 45, 47, 49, 63, 67, 83, 99, 125$. The size of the circle is proportional to $\log \Delta_K$.

REFERENCES

[1] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. 2002. PRIMES is in P. *Ann. of Math* 2 (2002), 781–793.

[2] Koen de Boer. 2016. *Computing the Power Residue Symbol.* Master's thesis. Nijmegen, Radboud University. www.koendeboer.com/masterthesis_deBoer.pdf

[3] Koen de Boer and Carlo Pagano. 2018. Calculating the Power Residue Symbol and Ibeta, Applications of Computing the Group Structure of the Principal Units of a p-adic Number Field Completion. (2018). www.koendeboer.com/calculating_the_power_residue_symbol_and_ibeta.pdf

[4] Wieb Bosma, John Cannon, and Catherine Playoust. 1997. The Magma algebra system. I. The user language. *J. Symbolic Comput.* 24, 3-4 (1997), 235–265. DOI: http://dx.doi.org/10.1006/jsco.1996.0125 Computational algebra and number theory (London, 1993).

[5] Jan Bouw. 2016. *On the computation of Hilbert norm residue symbols.* Ph.D. Dissertation. Universiteit Leiden.

[6] Henri Cohen. 1993. *A Course in Computational Algebraic Number Theory.* Springer-Verlag New York, Inc., New York, NY, USA.

[7] Henri Cohen. 2000. *Advanced topics in computational number theory.* Springer, New York, N.Y. and Berlin, Heidelberg. http://opac.inria.fr/record=b1096070

[8] Mario Daberkow. 2001. On Computations in Kummer Extensions. *J. Symb. Comput.* 31, 1/2 (2001), 113–131. DOI: http://dx.doi.org/10.1006/jsco.2000.1013

[9] N. Koblitz. 1977. *P-adic numbers, p-adic analysis, and zeta-functions.* Springer-Verlag. https://books.google.nl/books?id=SfruAAAAMAAJ

[10] H. Koch. 1997. *Algebraic Number Theory.* Springer Berlin Heidelberg. https://books.google.nl/books?id=JihjOAE0ldgC

[11] Michiel Kosters. 2014. Calculating norm residue symbols in polynomial time. (9 2014). Unpublished notes.

[12] F. Lemmermeyer. 2000. *Reciprocity Laws: From Euler to Eisenstein.* Springer. https://books.google.nl/books?id=EwjpPeK6GpEC

[13] Arjen K. Lenstra and Hendrik W. Lenstra Jr. (Eds.). 1993. *The development of the number field sieve.* Lecture Notes in Mathematics, Vol. 1554. Springer-Verlag, Berlin.

[14] A. K. Lenstra, H. W. Lenstra Jr., and László Lovász. 1982. Factoring polynomials with rational coefficients. *Math. Ann.* 261 (1982), 515–534.

[15] H.W. Lenstra Jr. 1995. Computing Jacobi Symbols in ALgebraic Number Fields. (1995). http://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1995g/art.pdf

[16] W. Narkiewicz. 2004. *Elementary and Analytic Theory of Algebraic Numbers* (3 ed.). Springer-Verlag Berlin Heidelberg. DOI: http://dx.doi.org/10.1007/978-3-662-07001-7

[17] J. Neukirch. 1999. *Algebraic Number Theory.* Springer Berlin Heidelberg. https://books.google.nl/books?id=plE3PwAACAAJ

[18] Phong Q. Nguyen and Brigitte Vallée (Eds.). 2010. *The LLL algorithm : survey and applications.* Springer, Berlin, Heidelberg. http://opac.inria.fr/record=b1130525

[19] M. Overholt. 2015. *A Course in Analytic Number Theory.* American Mathematical Society. https://books.google.nl/books?id=m2CuoQEACAAJ

[20] Carlo Pagano. 2017. Jump sets for local fields. (2017), Forthcoming.

[21] Carl Pomerance. 1996. A tale of two sieves. *Notices Amer. Math. Soc* 43 (1996), 1473–1485.

[22] J.-P. Serre. 1978. Une "formule de masse" pour les extensions totalement ramifies de degr donn d'un corps local. (1978).

[23] Douglas Squirrel. 2012. An algorithm for the power residue symbol. (2012). http://douglassquirrel.com/research/AnAlgorithmForThePowerResidueSymbol.pdf

[24] P. Stevenhagen. 2008. The arithmetic of number rings. *Algorithmic number theory* 44 (2008). http://www.math.leidenuniv.nl/~psh/ANTproc/08psh.pdf