Algorithms for ray class groups and Hilbert class fields*

Kirsten Eisenträger† Sean Hallgren‡

Abstract

This paper analyzes the complexity of problems from class field theory. Class field theory can be used to show the existence of infinite families of number fields with constant root discriminant. Such families have been proposed for use in lattice-based cryptography and for constructing error-correcting codes. Little is known about the complexity of computing them. We show that computing the ray class group and computing certain subfields of Hilbert class fields efficiently reduce to known computationally difficult problems. These include computing the unit group and class group, the principal ideal problem, factoring, and discrete log. As a consequence, efficient quantum algorithms for these problems exist in constant degree number fields.

1 Introduction

The central objects studied in algebraic number theory are number fields, which are finite extensions of the rational numbers $\mathbb{Q}$. Class field theory focuses on special field extensions of a given number field K. It can be used to show the existence of infinite families of number fields with constant root discriminant. Such number fields have recently been proposed for applications in cryptography and error correcting codes. In this paper we give algorithms for computing some of the objects required to compute such extensions of number fields. Similar to the approach in computational group theory [BBS09] where there are certain subproblems such as discrete log that are computationally difficult to solve, we identify the subproblems and show that they are the only obstacles. Furthermore, there are quantum algorithms for these subproblems, resulting in efficient quantum algorithms for constant degree number fields for the problems we study.

In class field theory the main problem is computing extensions of number fields: given a number K field, compute a field extension $L \supset K$ with certain properties. A central extension studied in class field theory is the Hilbert class field H of K. This extension field has very nice properties. For example, H has the same root discriminant as K, and it is the maximal abelian unramified extension of K. Class field theory also describes the subfields of the Hilbert class field and for these the root discriminant is constant as well. A related more general type of extension is the ray class field of K and its subfields. These extensions also have limited ramification and bounded root discriminant.

In addition to their central nature in number theory, such extensions have also been shown to have applications in computer science. Peikert and Rosen [PR07] have proposed using families of number fields for lattice-based cryptography. The ideals inside the ring of integers of a number

*This work was supported in part by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-08-1-0298.

†Department of Mathematics, Penn State University, eisentra@math.psu.edu. Partially supported by National Science Foundation grant DMS-0801123 and a Sloan Research Fellowship.

‡Department of Computer Science and Engineering, Penn State University, hallgren@cse.psu.edu. Partially supported by National Science Foundation grant CCF-0747274.

field are special types of lattices. Peikert and Rosen showed that if a family of number fields with constant root discriminant is used, then the lattices arising from this family allow a very good connection factor in a worst-case to average-case reduction between lattice problems. The main open problem they give is how to compute such families of number fields. Similar number fields also appeared in an alternate construction of error-correcting codes by Guruswami [Gur03].

Computing families of number fields with bounded root discriminant appears to be very difficult. The Golod-Shafarevich theorem [Roq67] implies that infinite families of number fields with constant root discriminant exist. That they are computable via Kummer thoery follows from algorithms in the two books by Cohen [Coh93, Coh00]. More recent developments appear in [BS08]. However, this is a relatively recent field and little is known about the complexity of computing these extensions. The focus has been on developing the fundamental algorithms, with the additional requirement that they be practical and that small examples can be computed. Running times are rarely given, and at times objects that are exponential size are used. Our focus will be different since we will require algorithms with asymptotically bounded running times.

To show the existence of infinite families of number fields with constant root discriminant as described above one constructs an infinite tower of Hilbert class fields. This is done by computing a tower of field extensions $H_0 \subset H_1 \subset H_2 \subset \cdots$, starting with a carefully chosen base field H_0, and, at each step, computing the Hilbert class field H_{i+1} of H_i. The Golod-Shafarevich theorem [Roq67] gives sufficient conditions on the base field H_0 to ensure that this tower is infinite.

In this paper our goal is to understand the complexity of two problems related to computing number fields of bounded root discriminant. In the first problem a number field K is given, and the goal is to compute an extension contained in the Hilbert class field of K. The second is to compute the ray class group of a given number field K. We give efficient reductions from these problems to a set of problems which are believed to be hard classically. These include factoring integers, computing discrete logs, computing the unit group of a number field, computing the class group of a number field, solving the principal ideal problem in a number field and factoring ideals. One reason to do this is that it clearly identifies the obstacles to finding efficient algorithms. Furthermore, in certain instances the problems in the reduction may not be as difficult as the instances used in cryptography. The second reason for this approach is that, with the exception of ideal factorization, there are already known efficient quantum algorithms for these problems when the degree of the number field is constant. We show that ideal factorization reduces to integer factorization, and therefore there is an efficient quantum algorithm for this problem and efficient quantum algorithms for ray class groups and certain subfields of the Hilbert class field in the constant degree case. There are no complexity theoretic results ruling out quantum algorithms for the arbitrary degree case since the problems are known to be in NP$\cap$coNP [Thi95].

There are two parameters for number fields, the degree and the discriminant, and the degree of the input number field is particularly difficult to deal with. Problems such as computing the unit group of an arbitrary degree number field are exponential in the degree, even for current quantum algorithms. One potential way to handle this obstacle is to use the more general ray class fields. These field extensions have the right requirements for some of the applications, but are more flexible than Hilbert class field towers, which are determined completely by the base field. Ray class fields have an extra parameter $\mathfrak{m}$ that allows different field extensions to be computed at every level of the tower. The extensions are tied to the ray class group $\mathrm{Cl}_{\mathfrak{m}}(K)$. This approach gives a wider choice in how to construct a tower, and perhaps the extra choice can help in making the computations more efficient. We give an efficient quantum algorithm for computing ray class groups which are necessary for computing ray class fields.

Approach and open problems. The algorithm for the ray class group $\mathrm{Cl}_{\mathfrak{m}}(K)$, given a number field K and a modulus $\mathfrak{m}$, follows the algorithm described in [Coh00] (see also [CDO98,

CDO01]) and computes a group extension from the following four-term right-exact sequence:

$$U(K) \xrightarrow{\rho} (\mathcal{O}_K/\mathfrak{m})^* \xrightarrow{\psi} \mathrm{Cl}_{\mathfrak{m}}(K) \xrightarrow{\phi} \mathrm{Cl}(K) \to 1.$$

To compute $\mathrm{Cl}_{\mathfrak{m}}(K)$ from this exact sequence we have to have generators and relations for the other three groups appearing in the exact sequence and we have to show that the three maps ρ, ψ and ϕ and their inverses are efficiently computable. Since we are showing reductions to computing the unit group $U(K)$ and class group $\mathrm{Cl}(K)$ (among other problems), we show that $(\mathcal{O}_K/\mathfrak{m})^*$ and the three maps can all be efficiently computed, and then a group extension algorithm can be used for $\mathrm{Cl}_{\mathfrak{m}}(K)$. As far as quantum algorithms are concerned, it does not seem clear how to directly compute the ray class group without going through the exact sequence, as one would need a notion of reduced ideal in the ray class group.

For both the Hilbert class field and the ray class group we are required to change the exponential-size representation used [Coh00] for algebraic numbers to two different compact representations. In each of the cases where the representation changes to a compact one we must show that we can still compute the same output even though we only have access to the compact representations, which are polynomial-size representations of the same object. For the Hilbert class field we show that compact representations of virtual units can be computed and that the field extension can be computed efficiently using these representations. For the ray class group we show that the maps and relations can be computed efficiently using these representations. We also change to a different representation for computing modulo an ideal, instead of the heuristic using the LLL algorithm from [Coh00] in order to bound the running time and prove correctness. We must analyze other algorithms such as factoring ideals and computing valuations. We show that factoring ideals reduces to factoring integers. We also show that for ψ, there is a way to efficiently map elements into $\mathrm{Cl}_{\mathfrak{m}}(K)$ using the closest vector problem in a lattice.

There are several open problems related to this work. One question is whether computing a family of number fields with constant root discriminant reduces to problems such as computing the unit group and class group in arbitrary degree number fields, or whether there exist constructions that avoid the degree. Another question is if there are quantum algorithms for computing the unit group, class group, and solving the principal ideal problem in arbitrary degree number fields. These problems are known to be in NP∩CoNP, which makes them good candidates for quantum algorithms. Another open problem is how to pass between various compact representations of algebraic numbers in arbitrary degree number fields. A related open question is whether the approximate shortest vector problem can be solved in ideal lattices. As pointed out in [PR07], these lattices have enough structure so that unlike the general case, the length of the shortest vector can be approximated. Perhaps the additional structure, using the fact that they are ideals and not just lattices would make them easier to solve by either a quantum or classical algorithm.

2 Background

The main objects we will compute in this paper are the ray class group $\mathrm{Cl}_{\mathfrak{m}}(K)$ of a number field K and modulus $\mathfrak{m}$ and certain subextensions of the Hilbert class field of K. To do this we will need to compute the unit group $U(K)$, the class group $\mathrm{Cl}(K)$, the group $(\mathcal{O}_K/\mathfrak{m})^*$, and maps between them. We start with the definition of a number field. A number field K is a finite extension of $\mathbb{Q}$. For references about the computational aspects see [Len92, Coh93, Thi95, Coh00, Hal05]. As a field extension, K can be generated by a single element θ in $\mathbb{C}$, $K = \mathbb{Q}(\theta)$. The element θ is a root of a monic irreducible polynomial of degree n with rational coefficients, which is called the minimal polynomial of θ. The number n is called the degree of K (over $\mathbb{Q}$) and an extension of

degree n is also an n-dimensional vector space over $\mathbb{Q}$. As a basis for the vector space of $K/\mathbb{Q}$ we can choose $1, \theta, \theta^2, ..., \theta^{n-1}$. There are different polynomial-time equivalent ways of describing the number field, such as an approximation to θ, or by giving θ's minimal polynomial.

First we define some quantities associated with number fields. A number field $K = \mathbb{Q}(\theta)$ of degree n has n embeddings into $\mathbb{C}$, which we will denote by K_i. If an embedding K_i is contained in $\mathbb{R}$ we call K_i a real embedding. Otherwise we refer to K_i as a complex embedding. Let s denote the number of real embeddings and t the number of pairs of complex conjugate embeddings. Then $n = s + 2t$. Let $m = s + t$. An element in K has n conjugates, and K has m absolute values, all of which correspond to the embeddings. Given an element $\alpha \in K$, $\alpha = \sum_{i=0}^{n-1} a_i \theta^i$ for some rational numbers $a_i \in \mathbb{Q}$, let $\alpha^{(j)}$ denote the jth conjugate of α, that is, $\alpha^{(j)} = \sum_{i=0}^{n-1} a_i \theta_j^i$. Here $\theta_1, \ldots, \theta_n$ are the roots of the minimal polynomial of θ. The jth absolute value $|\cdot|_j$ of a number α is a function of the absolute value in the jth conjugate field: $|\alpha|_j = |\alpha^{(j)}|$ if the embedding is real, and $|\alpha|_j = |\alpha^{(j)}|^2$ if the embedding is complex.

There are several problems associated to number fields which can be solved efficiently with a quantum algorithm [Hal05, SV05]. The first is computing the unit group, where a fundamental system of units for the finitely generated infinite abelian group can be computed. The second is computing the class group, which is a finite abelian group without unique group representatives. The third problem is the principal ideal problem, which computes a generator of a principal ideal. Classically, the best known algorithms take exponential time. The running times are in terms of the discriminant Δ and degree n of K.

The ring of integers is defined as the set of elements in K that are the root of some monic polynomial in $\mathbb{Z}[x]$. Computing $\mathcal{O}_K$ is polynomial-time equivalent to finding the largest square factor of a given positive integer [Chi89, BL94]. The unit group $\mathcal{O}_K^*$ is the set of invertible elements in $\mathcal{O}_K$. In the trivial case of $K = \mathbb{Q}$ and $\mathcal{O}_K = \mathbb{Z}$, $\mathcal{O}_K^* = \{\pm 1\}$. By Dirichlet's unit theorem, the unit group in general will be isomorphic to r copies of $\mathbb{Z}$, together with an efficiently (in constant dimension) classically computable root of unity. With a quantum algorithm it is possible to compute a fundamental system of units $\varepsilon_1, \ldots, \varepsilon_r$ that generate $\mathcal{O}_K^*$. This is not the entire story, yet, because these fundamental units may have exponentially many bits. Up to a root of unity μ, any $\varepsilon \in \mathcal{O}_K^*$ can be written as $\varepsilon = \mu \varepsilon_1^{k_1} \cdots \varepsilon_r^{k_r}$, where $k_1, \ldots, k_r \in \mathbb{Z}$.

Representation of elements. We now discuss how to represent K and $\mathcal{O}_K$, how to represent elements of K, and which computations with these elements are polynomial time. The standard representation of a number field K with its ring of integers $\mathcal{O}_K$ is to specify a $\mathbb{Z}$-basis $\omega_1, \ldots, \omega_n$ for $\mathcal{O}_K$, together with its multiplication table. That is, any number in $\alpha \in \mathcal{O}_K$ can be uniquely written as $\alpha = \sum_{i=1}^n a_i \omega_i$, where $a_1, \ldots, a_n \in \mathbb{Z}$. This basis will also be a $\mathbb{Q}$ basis for K, so any $\alpha \in K$ can be uniquely written as $\alpha = \sum_{i=1}^n a_i \omega_i$, where $a_1, \ldots, a_n \in \mathbb{Q}$. The multiplication table is a set of n^3 integers $(c_{ijk})_{ijk}$ that defines multiplication in K: $\omega_i \omega_j = \sum_{k=1}^n c_{ijk} \omega_k$. The discriminant Δ of K is the determinant of the matrix $(\mathrm{Tr}\ (\omega_i \omega_j))_{ij}$, where $\mathrm{Tr} : K \to \mathbb{Q}$ is the trace map. The multiplication table for $\mathcal{O}_K$ can be transformed so that it has $O(n^4(2 + \log|\Delta|))$ bits. For the purposes of analyzing running times it is customary to use Δ as the input, and an algorithm is polynomial time if it is polynomial in $\log|\Delta|$ and the degree n. In this representation, addition, multiplication, and division are polynomial time. However, representing a fundamental set of units for $\mathcal{O}_K^*$ is not polynomial size.

Compact representations allow a polynomial-size representation of the fundamental units. Generally speaking, a compact representation of a number $\alpha \in K$ is a set of numbers $\gamma_1, \ldots, \gamma_m \in K$ in the standard representation and $k_1, \ldots, k_m \in \mathbb{Z}$, where the total amount of data is polynomial size, and $\alpha = \Pi_{i=1}^m \gamma_i^{k_i}$. This representation is not unique. Elements in this compact representation can be added, multiplied, and divided provided that the number field is restricted to constant dimension. Another polynomial-size representation is from the Log embedding into $\mathbb{R}^r$, defined

by $\alpha \mapsto \mathrm{Log}\alpha = (\log|\alpha|_1, \ldots, \log|\alpha|_r)$. These are irrational numbers, but keeping them to some precision will suffice.

Fractional ideals. Another object we will need are the (fractional) ideals of $\mathcal{O}_K$. An ideal I of $\mathcal{O}_K$ is a subset of $\mathcal{O}_K$ that is closed under addition and under multiplication by elements of $\mathcal{O}_K$. An ideal has a $\mathbb{Z}$-basis $\beta_1, \ldots, \beta_n$ and is represented by an integer matrix $(a_{ij})_{ij} \in \mathbb{Z}^{n\times n}$ where $\beta_i = \sum_{j=0}^{n} a_{ij}\omega_j$. A fractional ideal I of K is a subset of K such that dI is an ideal for some integer d, and representing fractional ideals includes the extra parameter d. The matrix $(a_{ij})_{ij}$ representing the ideal can be kept in HNF form, and in this way representation of ideals is unique. It is possible to multiply two ideals in polynomial time, and to multiply an ideal by an element of K. There is an important class of ideals called reduced ideals for which it is possible to keep the representation size bounded by a polynomial.

Every (fractional) ideal I can be written in a unique way as $I = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(I)}$, where the product is over the set of prime ideals of $\mathcal{O}_K$, the exponents $v_{\mathfrak{p}}(I)$ are integers, and only finitely many exponents are nonzero. The exponent $v_{\mathfrak{p}}(I)$ is called the valuation of I at the prime ideal $\mathfrak{p}$. Two ideals $I, J \subseteq \mathcal{O}_K$ are called coprime, if $I+J = \mathcal{O}_K$, or equivalently if their factorizations into powers of prime ideals have no common factor $\mathfrak{p}$ that occurs with nonzero exponent in both factorizations. A fractional ideal $\mathfrak{a}$ is coprime to an ideal I if we can write $\mathfrak{a} = \mathfrak{b}/\mathfrak{c}$ with $\mathfrak{b}, \mathfrak{c}$ integral ideals which are coprime to I. We say that an element $\alpha \in K^*$ and an ideal I are coprime if the ideals (α) and I are coprime. For an ideal $I \subseteq \mathcal{O}_K$, we define the norm of I to be the cardinality of the finite quotient ring $\mathcal{O}_K/I$ and denote it by $\mathcal{N}(I)$.

The ray class group. Let $\mathfrak{m}$ be a modulus of a number field K, i.e. a pair $(\mathfrak{m}_0, \mathfrak{m}_\infty)$, where $\mathfrak{m}_0$ is an integral ideal of $\mathcal{O}_K$ and $\mathfrak{m}_\infty$ is a subset of the real embeddings of K into $\mathbb{C}$. Given an element $\alpha \in K^*$ we define $\alpha \equiv 1 \bmod {}^*\mathfrak{m}$ to mean that for all primes $\mathfrak{p}$ appearing in the factorization of $\mathfrak{m}_0$ we have $v_{\mathfrak{p}}(\alpha - 1) \geq v_{\mathfrak{p}}(\mathfrak{m}_0)$ and that $\sigma_i(\alpha) > 0$ for all embeddings $\sigma_i : K \hookrightarrow \mathbb{R}$ in $\mathfrak{m}_\infty$. The ray class group $\mathrm{Cl}_{\mathfrak{m}}(K)$ is defined as the set of ideals coprime to $\mathfrak{m}$ modulo the principal ideals coprime to $\mathfrak{m}$ that can be generated by an element α with $\alpha \equiv 1 \bmod {}^*\mathfrak{m}$.

Given a modulus $\mathfrak{m} = (\mathfrak{m}_0, \mathfrak{m}_\infty)$ as above, we define the group $(\mathcal{O}_K/\mathfrak{m})^*$ to be $(\mathcal{O}_K/\mathfrak{m})^* = (\mathcal{O}_K/\mathfrak{m}_0)^* \times \mathbb{F}_2^{|\mathfrak{m}_\infty|}$. There is a natural group homomorphism ρ from the elements $\alpha \in K$ which are coprime to $\mathfrak{m}$ into $(\mathcal{O}_K/\mathfrak{m})^*$. Given such an α, write $\alpha = \beta/\gamma$ with $\beta, \gamma \in \mathcal{O}_K$ and coprime to $\mathfrak{m}$. Now define $\rho(\alpha)$ to be $(\overline{\beta}/\overline{\gamma}, (\mathrm{sign}(\sigma_i(\alpha)_{\sigma_i \in \mathfrak{m}_\infty})))$. Here $(\mathrm{sign}(\sigma_i(\alpha)_{\sigma_i \in \mathfrak{m}_\infty})$ is the vector of signs of α under the embeddings in $\mathfrak{m}_\infty$. (These embeddings are all real-valued.) This vector is also referred to as the signature of α, and also denoted $s(\alpha)$.

Hilbert class fields. Finally, to define the Hilbert class field of a number field K we need to introduce the notion of an unramified extension. An extension L/K of number fields is unramified if any prime ideal $\mathfrak{p}$ of K factors in L as $\mathfrak{p}\mathcal{O}_L = \prod_{\mathfrak{q}} \mathfrak{q}^{e_{\mathfrak{q}}}$ with all exponents being 0 or 1 and if all real embeddings of K extend to real embeddings in L. An extension L/K is abelian if it is Galois with abelian Galois group. We define the Hilbert class field H of a number field K to be the maximal abelian unramified extension of K. It follows from class field theory that the Galois group of H over K is isomorphic to the class group of K, and K and H have the same root discriminant because the extension H/K is unramified.

3 Computing groups $(\mathcal{O}_K/I)^*$

Let I be an ideal of $\mathcal{O}_K$. In this section we show how to compute the finite abelian multiplicative group $(\mathcal{O}_K/I)^*$. In the special case where I is a prime ideal, $(\mathcal{O}_K/I)^*$ is the multiplicative group of a finite field, but for our algorithms we need the generalization to arbitrary ideals. This will be used to compute the ray class group, where a subset of the generators will come from $(\mathcal{O}_K/\mathfrak{m})^*$ for

a modulus $\mathfrak{m}$. It will also be used to compute Hilbert class fields.

We start by describing the algorithm for computing $(\mathcal{O}_K/I)^*$, then we describe how each of the steps work. The description will be in the context of $(\mathcal{O}_K/\mathfrak{m})^*$, which is slightly more general, since $\mathfrak{m}$ also includes a subset $\mathfrak{m}_\infty$ of the real embeddings. Recall that elements of $(\mathcal{O}_K/\mathfrak{m})^*$ are pairs $(\bar{\alpha}, w)$ where $\alpha \in \mathcal{O}_K$ is coprime to $\mathfrak{m}_0$ and $w \in \mathbb{F}_2^{|\mathfrak{m}_\infty|}$. So we have to compute $(\mathcal{O}_K/\mathfrak{m}_0)^*$ and the infinite part.

We change the representation of ideals used in [Coh00] for asymptotic efficiency. Let $\mathfrak{m}_0 = \Pi\, \mathfrak{p}_i^{v_i}$. We need generators and relations for $(\mathcal{O}_K/\mathfrak{m}_0)^*$ which is isomorphic to $(\mathcal{O}_K/\mathfrak{p}_1^{v_1})^* \times (\mathcal{O}_K/\mathfrak{p}_2^{v_2})^* \times \cdots \times (\mathcal{O}_K/\mathfrak{p}_\ell^{v_\ell})^*$ by the Chinese Remainder Theorem. Lemma 3.1 below shows how to compute generators for $(\mathcal{O}_K/\mathfrak{m}_0)^*$ from generators for $(\mathcal{O}_K/\mathfrak{p}^k)^*$ for a prime ideal $\mathfrak{p}$ and a positive integer k. In Section 3.1 we will show how to obtain generators for the groups $(\mathcal{O}_K/\mathfrak{p}^k)^*$.

Lemma 3.1. *(CRT for ideals) Let $\mathfrak{a}, \mathfrak{c}$ be coprime ideals of $\mathcal{O}_K$, and let $\mathfrak{b} = \mathfrak{a}\mathfrak{c}$. Assume that the decompositions of $(\mathcal{O}_K/\mathfrak{a})^*$ and $(\mathcal{O}_K/\mathfrak{c})^*$ are known. Assume that $(\mathcal{O}_K/\mathfrak{a})^* = \bigoplus(\mathbb{Z}/a_i\mathbb{Z})\overline{\alpha_i}$, with $\alpha_i \in \mathcal{O}_K$ and that $(\mathcal{O}_K/\mathfrak{c})^* = \bigoplus(\mathbb{Z}/c_j\mathbb{Z})\overline{\gamma_j}$ with $\gamma_j \in \mathcal{O}_K$.*

1. *We can find $a \in \mathfrak{a}, c \in \mathfrak{c}$ in time poly$(n, \log \Delta_K)$ such that $a + c = 1$.*
2. *We have*

$$(\mathcal{O}_K/\mathfrak{b})^* = \bigoplus(\mathbb{Z}/a_i\mathbb{Z})(\overline{c\alpha_i + a}) \oplus \bigoplus(\mathbb{Z}/c_j\mathbb{Z})(\overline{a\gamma_j + c}).$$

Proof. The claim is [Coh00], Lemma 4.2.1 (1). The second is [Coh00], Lemma 4.2.1 (3). □

This motivates the following algorithm.

Algorithm 1. Computing $(\mathcal{O}_K/\mathfrak{m})^*$
Input: K, $\mathcal{O}_K$, modulus $\mathfrak{m} = \mathfrak{m}_0\mathfrak{m}_\infty$
Output: Generators $\varepsilon_{\mathfrak{p},i}$ and relations

1. *Factor $\mathfrak{m}_0 = \Pi_\mathfrak{p}\mathfrak{p}^{v_\mathfrak{p}}$ using Algorithm 2 in Section 4.*
2. *For each prime $\mathfrak{p}$ in the factorization, compute generators $\delta_{\mathfrak{p},i}$ and integers $d_{\mathfrak{p},i}$ such that $(\mathcal{O}_K/\mathfrak{p}^{v_\mathfrak{p}})^* = \bigoplus_i(\mathbb{Z}/d_{\mathfrak{p},i}\mathbb{Z})\overline{\delta_{\mathfrak{p},i}}$. The generators $\delta_{\mathfrak{p},i}$ will be coprime to $\mathfrak{p}$.*
3. *Adjust each generator so that it is coprime to $\mathfrak{m}_0$: compute $a_\mathfrak{p} \in \mathfrak{p}^{v_\mathfrak{p}}$ and $c_\mathfrak{p} \in \mathfrak{m}_0/\mathfrak{p}^{v_\mathfrak{p}}$ whose sum is 1, and set $\varepsilon_{\mathfrak{p},i} = (\overline{c_\mathfrak{p}\delta_{\mathfrak{p},i} + a_\mathfrak{p}}, 0)$.*
4. *Add generators e_j for the infinite part $\mathbb{F}_2^{|\mathfrak{m}_\infty|}$, where e_j are standard basis vectors $(1 \le j \le |\mathfrak{m}_\infty|)$.*
5. *The relations matrix is a diagonal matrix with $d_{\mathfrak{p},i}$ the entry for generator $\varepsilon_{\mathfrak{p},i}$.*

Theorem 3.2. *Given a number field K, $\mathcal{O}_K$, and a modulus $\mathfrak{m}$, computing the group $(\mathcal{O}_K/\mathfrak{m})^*$ reduces to factoring $\mathfrak{m}$ and discrete log in finite groups in time poly$(n, \log\Delta, \log(\mathcal{N}(\mathfrak{m}_0)))$.*

Proof. Computing $(\mathcal{O}_K/\mathfrak{p}^k)^*$ can by done by Lemma 3.5. By definition of $(\mathcal{O}_K/\mathfrak{p}^k)^*$, generators for this group are coprime to $\mathfrak{p}$. These generators can be reduced modulo the ideal $\mathfrak{m}_0$, and coprimeness is preserved by Proposition 3.3 below. Elements $a_\mathfrak{p}, c_\mathfrak{p}$ as in Algorithm 1 can be computed in polynomial time by Lemma 3.1 and computing the linear combination is efficient. Steps 4 and 5 are trivial. □

Proposition 3.3. *Suppose $I \subseteq \mathcal{O}_K$ is an ideal and $a \in \mathcal{O}_K$ is coprime to I. Let b be an element of $\mathcal{O}_K$ with $a \equiv b \pmod{I}$. Then b is coprime to I.*

Proof. Let $I = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}}$ for $v_{\mathfrak{p}} > 0$ be the factorization of I into powers of prime ideals. We have $a \equiv b \pmod{I}$, so $b = a + \alpha$ for some element $\alpha \in I$. Since $\alpha \in I$, $(\alpha) \subseteq I$, and hence $0 \leq v_{\mathfrak{p}}(I) \leq v_{\mathfrak{p}}(\alpha)$ for all prime ideals $\mathfrak{p}$ of $\mathcal{O}_K$ ([Coh93, p.193]). Hence for all prime ideals $\mathfrak{p}$ appearing in the above factorization of I we have $v_{\mathfrak{p}}(b) = v_{\mathfrak{p}}(a+\alpha) = 0$, by standard properties of valuations since $v_{\mathfrak{p}}(a) = 0$ and $v_{\mathfrak{p}}(\alpha) \geq 1$. Hence the factorizations of (b) and I have no prime ideals in common, and so b and I are relatively prime. □

3.1 Computing $(\mathcal{O}_K/\mathfrak{p}^k)^*$.

In this section we show how to compute $(\mathcal{O}_K/\mathfrak{p}^k)^*$, which is used for computing $(\mathcal{O}_K/\mathfrak{m})^*$ the ray class group and Hilbert class fields. First we need to define the group $(1+\mathfrak{a})/(1+\mathfrak{b})$.

Definition 3.4. *Let K be a number field. Let $\mathfrak{a}$ and $\mathfrak{b}$ be two nonzero ideals of $\mathcal{O}_K$ with $\mathfrak{a} \mid \mathfrak{b} \mid \mathfrak{a}^\ell$ for some integer ℓ. We denote by $(1+\mathfrak{a})/(1+\mathfrak{b})$ the quotient of the multiplicative set $1+\mathfrak{a}$ by the equivalence relation R given by $(1+x)R(1+y)$ if and only if $x \equiv y \mod \mathfrak{b}$. Then multiplication in K induces multiplication $(1+\mathfrak{a})/(1+\mathfrak{b})$ and makes the set $(1+\mathfrak{a})/(1+\mathfrak{b})$ into an abelian group.*

Proof. This is Definition and Proposition 1.3, page 776, in [CDO98]. □

Lemma 3.5. *Given a prime ideal $\mathfrak{p}$ of $\mathcal{O}_K$, $\mathcal{O}_K$, and an integer k, computing the group $(\mathcal{O}_K/\mathfrak{p}^k)^*$ reduces to computing discrete log in finite groups in time poly$(n, \log \Delta_K, \log \mathcal{N}(\mathfrak{p}), k)$.*

Proof. Following the outline of the algorithm in [Coh00], we analyze the running time and prove correctness below. The computationally difficult parts are order finding in $(\mathcal{O}_K/\mathfrak{p})^*$ and discrete log in the $(1+\mathfrak{p})/(1+\mathfrak{p}^k)$ groups.

The approach is based on the following proposition which decomposes $(\mathcal{O}_K/\mathfrak{p}^k)^*$ into a product of two groups:

Proposition 3.6. *Let $\mathfrak{p}$ be a prime ideal of $\mathcal{O}_K$ and let p be a prime such that $q = p^f$ is the cardinality of the residue field $\mathcal{O}_K/\mathfrak{p}$. Let*

$$W = \{x \in (\mathcal{O}_K/\mathfrak{p}^k)^* : x^{q-1} = 1\}$$

and

$$G_{\mathfrak{p}} = (1+\mathfrak{p})/(1+\mathfrak{p}^k).$$

Then $(\mathcal{O}_K/\mathfrak{p}^k)^* = W \times G_{\mathfrak{p}}$.

First generators are computed for each factor. Compute generators for $(1+\mathfrak{p}^a)/(1+\mathfrak{p}^{2a})$ for increasing values of a from 1 up to (around) k. These can be computed from generators for $\mathfrak{p}$, by taking powers and adding 1. Next compute an element g_0 of order $q-1$ of $(\mathcal{O}_K/\mathfrak{p})^*$ by randomly selecting an element and computing the order in that group to test if it is $q-1$. Then Hensel lifting can be applied to lift g_0 to a generator of W in polynomial time.

Next relations must be computed for these generators. This will require computing discrete logs in $(1+\mathfrak{p})/(1+\mathfrak{p}^k)$ with respect to the set of generators from all $(1+\mathfrak{p}^a)/(1+\mathfrak{p}^{2a})$ groups taken mod $(1+\mathfrak{p}^k)$ to compute the relations. □

4 Factoring ideals and computing valuations

In this section we show that factoring ideals in number fields reduces to factoring integers and that computing valuations in any number field is polynomial time in $\log(\Delta_K)$ and n. Factoring an ideal I can be done by first computing a set of potential prime ideals that may divide the ideal I, and then computing valuations to see which ones occur in the factorization. It is sufficient to compute the set of prime ideals above each prime integer p dividing the norm of I. For a prime integer p, the prime ideals above p are the prime ideals of $\mathcal{O}_K$ which contain $p\mathcal{O}_K$. They are exactly the prime ideals which occur in the factorization of $p\mathcal{O}_K$. Obtaining them is the main computational step in the algorithm. The outline of the ideal factorization algorithm is described in [Coh00] as Algorithm 2.3.22. Our aim is to analyze the running time of this algorithm and show that factoring is the only computationally difficult part.

Algorithm 2. Ideal factorization
Input: Number field K, fractional ideal $I = (d, A)$ of K.
Output: Prime ideals $\mathfrak{p}$, integers $v_\mathfrak{p} = v_\mathfrak{p}(I)$ such that $I = \Pi \mathfrak{p}^{v_\mathfrak{p}}$.

1. *Compute the norm N of the integral ideal dI.*
2. *Factor N as $N = \Pi p^{e_p}$, with $e_p > 0$.*
3. *For each prime p dividing N, compute the prime ideals $\mathfrak{p}_1, ..., \mathfrak{p}_k$ above p.*
4. *For each prime p dividing N, and each $\mathfrak{p} \supset p\mathcal{O}_K$ from Step (3), compute $v_\mathfrak{p}(dI)$, giving the exponent of $\mathfrak{p}$ in the factorization of dI.*
5. *For each $\mathfrak{p}$ found with nonzero valuation, output $\mathfrak{p}, v_\mathfrak{p}(dI)$. We have $dI = \Pi\ \mathfrak{p}^{v_\mathfrak{p}(dI)}$.*
6. *Repeat steps (1)–(5) for the integral ideal $d\mathcal{O}_K$, then subtract the exponents of $d\mathcal{O}_K$ from the exponents computed above for the ideal dI for each prime, giving the primes $\mathfrak{p}$ and the exponents $v_\mathfrak{p}$ such that $I = \prod \mathfrak{p}^{v_\mathfrak{p}}$.*

Lemma 4.1. *Factoring fractional ideals of a number field K into a product of prime ideals of $\mathcal{O}_K$ reduces to factoring integers in polynomial time.*

Proof. First compute the ring of integers $\mathcal{O}_K$, which reduces to factoring [Chi89, BL94]. If an integral ideal J is given by its HNF (or by any matrix A) on a basis of $\mathcal{O}_K$, then the norm of J is the absolute value of the determinant of A. (See [Coh93, Proposition 4.7.4].)

In Section 4.1 we show that computing the primes above p can be done in polynomial time and in Section 4.2 we show that computing valuations can be done in polynomial time. □

4.1 Computing the primes above p.

Let K be a number field which is generated over $\mathbb{Q}$ by an algebraic integer θ. For a prime integer p, computing the primes above p in the special case when $p \nmid [\mathcal{O}_K : \mathbb{Z}[\theta]]$ can be done using the well-known theorem below. We first describe this case and then give an algorithm that works in the general case. Note that Lenstra [Len92, Theorem 4.9] gives a sketch of a different algorithm that also works for orders.

Theorem 4.2. *Let K be a number field of degree n, generated over $\mathbb{Q}$ by an algebraic integer θ. Let $f(X)$ be the minimal polynomial of θ. Let p be a prime such that $p \nmid [\mathcal{O}_K : \mathbb{Z}[\theta]]$. Consider the factorization $f(X) = \Pi f_i(X)^{e_i}$ into irreducible factors over $\mathbb{F}_p$. Then $p\mathcal{O}_K = \Pi \mathfrak{p}_i^{e_i}$ where $\mathfrak{p}_i = p\mathcal{O}_K + f_i(\theta)\mathcal{O}_K$.*

Proof. This is Theorem 27 in [Mar77, Chapter 3]. □

Therefore, factoring $p\mathcal{O}_K$ when $p \nmid [\mathcal{O}_K : \mathbb{Z}[\theta]]$ now works as follows. First compute θ and f, and factor f, each of which is polynomial time [Rab80, Len92]. Next compute the desired prime ideal $\mathfrak{p}_i$ using the identity $\mathfrak{p}_i = p\mathcal{O}_K + f_i(\theta)\mathcal{O}_K$.

Next we describe a general algorithm that does not require any assumptions on p. The approach is as follows. If an ideal I is a product of distinct prime ideals, then $\mathcal{O}_K/I$ is a product of fields, and we can decompose it by finding idempotents. When all the prime ideals appearing in the factorization of I are primes above p then $\mathcal{O}_K/I$ is an $\mathbb{F}_p$-algebra, and non-trivial idempotents can be found efficiently. The algorithm below first computes the radical I_p of $p\mathcal{O}_K$, which is the product of all distinct prime ideals above p, and then decomposes $\mathcal{O}_K/I_p$. We next describe this in more detail.

Definition 4.3. *Let I be a proper ideal in a commutative ring R (with identity). The* radical *or* nilradical *of I is the intersection of all prime ideals of R containing I. The radical of the zero ideal, i.e. the intersection of all prime ideals of R, is often referred to as the nilradical of the ring R.*

The following theorem gives a different description that will allow computation of the radical.

Theorem 4.4. *Let I be an ideal in a commutative ring R. The radical of I equals*

$$\{r \in R : r^n \in I \text{ for some } n \in \mathbb{Z}^+\}.$$

Proof. This is Theorem 2.6 in [Hun80, p. 379]. □

Now let $A = \mathcal{O}_K/I_p$, where I_p is the radical of $p\mathcal{O}_K$. By the above theorem, together with the fact that in rings of integers the intersection of distinct prime ideals equals their product, we have

$$\begin{aligned} I_p &= \{x \in \mathcal{O}_K : x^m \in p\mathcal{O}_K \text{ for some } m \in \mathbb{Z}^+\} \\ &= \prod_{\mathfrak{p} \supseteq p\mathcal{O}_K} \mathfrak{p}, \end{aligned}$$

and the product is over all distinct primes lying above p. Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_m$ be the distinct prime ideals of $\mathcal{O}_K$ above p. By the Chinese Remainder Theorem, A is isomorphic to $\mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_m$, so A is a product of finite fields and also an $\mathbb{F}_p$-algebra.

We can now give the algorithm to compute the primes above any prime integer p. In the first part of the algorithm we compute a $\mathbb{Z}$-basis of I_p. For this we first compute an $\mathbb{F}_p$-basis for $I_p/p\mathcal{O}_K$ by using the fact that the nilradical of the finite ring $\mathcal{O}_K/p\mathcal{O}_k$ is $I_p/p\mathcal{O}_K$. From this we can easily obtain a generating set and then a basis of I_p.

Algorithm 3. Primes above p
Input: Number field K, $\mathcal{O}_K$, prime p.
Output: The set of prime ideals above p.

1. *Compute a $\mathbb{Z}$-basis of I_p.*

   (a) *Compute an $\mathbb{F}_p$-basis of $I_p/p\mathcal{O}_K$.*

   i. Choose q to be a power of p larger than n.

   ii. Consider the homomorphism $\mathcal{O}_K/p\mathcal{O}_K \to \mathcal{O}_K/p\mathcal{O}_K$ defined by $x \mapsto x^q$. Compute the matrix $A \in \mathbb{F}_p^{n \times n}$ of this homomorphism by solving equations mapping the basis $\{\omega_1, ..., \omega_n\}$ to $\{\omega_1^q, ..., \omega_n^q\}$.

   iii. Compute $\ker A$*, which identifies the nilradical $I_p/p\mathcal{O}_K$ of $\mathcal{O}_K/p\mathcal{O}_K$.*

(b) *Compute a basis of I_p from generators consisting of pullbacks of the generators of $I_p/p\mathcal{O}_K$ together with the basis $p\omega_1, ..., p\omega_n$ of $p\mathcal{O}_K$.*

2. *Given an ideal I that is a product of distinct prime ideals lying above a prime p, this step gives a nontrivial factorization if one exists or specifies it is prime. Repeating this factors I completely.*

   (a) *Compute an idempotent $\bar{e} \in \mathcal{O}_K/I$, and let e be any lift to $\mathcal{O}_K$.*

   (b) *If a non-trivial idempotent was found, compute $H_1 = I + e\mathcal{O}_K$ and $H_2 = I + (1-e)\mathcal{O}_K$, giving a nontrivial factorization $I = H_1H_2$. Otherwise I is prime.*

Proposition 4.5. *Given a number field K, $\mathcal{O}_K$, and a prime p, Algorithm 3 computes the primes above p in polynomial time.*

Proof. The map $x \mapsto x^q$ is a power of the pth power Frobenius map, which is a homomorphism since $\mathcal{O}_K/p\mathcal{O}_K$ is an $\mathbb{F}_p$-algebra. So step 1 requires solving linear equations over finite fields and computing the kernel of matrix. By [Coh93, Lemma 6.16] the kernel of A is the desired nilradical. For step 2, the an idemptotent in this case can be computed efficiently and a factorization of I results by by [Coh93, Proposition 6.2.8] and the following discussion. The $\mathbb{Z}$-basis for I_p can be computed in polynomial time. □

4.2 Computing valuations in any number field.

Here we analyze the running time of the algorithm in [Coh93] for computing valuations. For K and $\mathcal{O}_K$, this algorithm takes a prime ideal $\mathfrak{p}$ and an ideal I and computes $v_{\mathfrak{p}}(I)$. The steps are to compute $a \in K - \mathcal{O}_K$ such that $a\mathfrak{p} \subset \mathcal{O}_K$, and then to compute the largest $v \in \mathbb{Z}$ such that $a^v I \subset \mathcal{O}_K$. To compute a, we use the fact that $a = \beta/p$, where $p \in \mathfrak{p}$ (p is on the diagonal of the HNF of $\mathfrak{p}$) and we set up a systems of linear equations to solve mod p to get β. We need mn equations in n indeterminates for $\beta \in \mathcal{O}_K$, $\beta \in \mathcal{O}_K - p\mathcal{O}_K$, and $\beta\mathfrak{p} \subset p\mathcal{O}_K$. After we have a we try each possible exponent v up to $O(n \cdot \log \mathcal{N}(I))$, each time checking whether or not $\mathcal{N}(a^v I) = \mathcal{N}(a\mathcal{O}_K)^v \mathcal{N}(I) \in \mathbb{Z}$. So computing valuations is polynomial time also in the degree.

The following proposition allows us to compute valuations:

Proposition 4.6. *Let $\mathfrak{p}$ be a prime ideal of a number field K, and let I be an integral ideal of the ring of integers $\mathcal{O}_K$. There exists $a \in K - \mathcal{O}_K$ such that $a\mathfrak{p} \subset \mathcal{O}_K$. The valuation $v_{\mathfrak{p}}(I)$ of I is the largest integer v such that $a^v I \subset \mathcal{O}_K$.*

Proof. The existence of the element a follows from Lemma 2 in [Mar77, Chapter 2]. The second part of the claim is Proposition 9.1 in [Ste08]. □

4.2.1 Computing a suitable element $a \in K - \mathcal{O}_K$.

Lemma 4.7. *Let K be a number field and $\mathfrak{p}$ a prime ideal such that $\mathcal{O}_K/\mathfrak{p}$ has characteristic p. Let $\omega_1, \ldots, \omega_n$ be a $\mathbb{Z}$-basis for $\mathcal{O}_K$, and let $\gamma_1, \ldots, \gamma_m$ be generators of $\mathfrak{p}$ which are given in terms of the $\mathbb{Z}$-basis. Then we can compute an element a as in Proposition 4.6 in polynomial time.*

Proof. This proof analyzes the running time of the algorithm described in [Coh93, p. 202]. To find an element a as in the lemma proceed as follows: The condition $a\mathfrak{p} \subset \mathcal{O}_K$ implies that $ap \in \mathcal{O}_K$, so $a = \beta/p$ with $\beta \in \mathcal{O}_K$. Since $a \in K - \mathcal{O}_K$, we must have $\beta \in \mathcal{O}_K - p\mathcal{O}_K$. Also $\beta\mathfrak{p} = (ap)\mathfrak{p} = p(a\mathfrak{p}) \subset p\mathcal{O}_K$.

Since $\beta \in \mathcal{O}_K$, we can write β as

$$\beta = \sum_{i=1}^{n} x_i \omega_i \quad \text{with } x_i \in \mathbb{Z}.$$

The condition that $\beta \in \mathcal{O}_K - p\mathcal{O}_K$ is equivalent to saying that not all x_i are divisible by p. Since $\beta\mathfrak{p} \subset p\mathcal{O}_K$, we have that $\beta\gamma_j$ is divisible by p, $j = 1, \ldots, m$. Hence

$$\beta\gamma_j = \left(\sum_i x_i \omega_i \right) \gamma_j = p \cdot \delta_j$$

for some $\delta_j \in \mathcal{O}_K$ $(j = 1, \ldots, m)$. Let

$$\omega_i \gamma_j = \sum_{1 \le k \le n} a_{i,j,k} \omega_k.$$

We have

$$\begin{aligned} \beta\gamma_j &= \left(\sum_{i=1}^{n} x_i \omega_i \right) \gamma_j \\ &= \sum_{i=1}^{n} x_i (\omega_i \gamma_j) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{n} a_{i,j,k} \omega_k \\ &= \sum_{k=1}^{n} \omega_k \sum_{i=1}^{n} x_i a_{i,j,k}. \end{aligned}$$

Since the ω_k's form a $\mathbb{Z}$-basis of $\mathcal{O}_K$ and since $\beta\gamma_j$ is divisible by p, this implies that we get the following system of $n \cdot m$ equations: For all $j \in \{1, \ldots, m\}$ and $k \in \{1, \ldots, n\}$ we have

$$\sum_{i=1}^{n} a_{i,j,k} x_i \equiv 0 \mod p.$$

This is a system of mn equations in n indeterminates $x_1, \ldots, x_n$ that we want to solve in $\mathbb{Z}/p\mathbb{Z}$. (If $\beta \in \mathcal{O}_K$ has the right properties, then clearly $\beta + p\omega_i$ works as well, so it is enough to determine the coefficients x_i of β modulo p.) Since β is not divisible by p in $\mathcal{O}_K$, this is the same as looking for a nontrivial solution $x_1, \ldots, x_n$ in $\mathbb{Z}/p\mathbb{Z}$. □

4.2.2 Computing $v_{\mathfrak{p}}(I)$.

Proposition 4.8. *Let K be a number field with ring of integers $\mathcal{O}_K$. Let I be an ideal of $\mathcal{O}_K$ in HNF. Let $\omega_1, \ldots, \omega_n$ be a $\mathbb{Z}$-basis of $\mathcal{O}_K$, and let $\mathfrak{p}$ be a prime ideal with generators $\gamma_1, \ldots, \gamma_m$. The valuation $v_{\mathfrak{p}}(I)$ can be computed in polynomial time.*

Proof. By Proposition 4.6 we have to compute an element $a \in K - \mathcal{O}_K$ as in the proposition and we have to compute the largest integer v such that $a^v I \subset \mathcal{O}_K$. By Lemma 4.7 we can compute the element a in polynomial time. To find the value of v, we compute the norm $\mathcal{N}(a^k I) = \mathcal{N}(a^k \mathcal{O}_K)\mathcal{N}(I)$ for different values of k. Since a fractional ideal is integral if and only if its norm is

an integer, we want the largest v such that $\mathcal{N}(a^v I) \in \mathbb{Z}$. This v can be determined as follows: we can compute the norm of I and the norm of $a\mathcal{O}_K$. Then $\mathcal{N}(a^k\mathcal{O}_K) = \mathcal{N}(a\mathcal{O}_K)^k$. Let $\mathcal{N}(I) = \prod p_i^{e_i}$ with $e_i > 0$. Let $\mathcal{N}(a\mathcal{O}_K) = \prod q_i^{a_i}$, with only primes q_i appearing for which $a_i \neq 0$. Since $\mathcal{N}(a\mathcal{O}_K) \in \mathbb{Q} - \mathbb{Z}$, some a_i must be less than zero. Then $0 \leq v \leq n \cdot \max e_i \leq n \cdot \log \mathcal{N}(I)$, where $n = [K : \mathbb{Q}]$, so we can just test the possible values and compute the norm each time. □

5 Computing subfields of Hilbert class fields

In this section we show how to compute subfields of Hilbert class fields of prime degree ℓ when the field contains a primitive ℓth root of unity. Class field theory tells us that the Galois group of the Hilbert class field L of K is isomorphic to the class group $\mathrm{Cl}(K)$ of K. By Galois theory this implies a that there is a one-to-one correspondence between subgroups of $\mathrm{Cl}(K)$ and subfields of L containing K. Since the class group can have exponential size it is not possible in general to have an efficient algorithm for the Hilbert class field in terms of the input size. Therefore we restrict to subfields of smaller size. The approach below constructs the desired subfield of the Hilbert class field via Kummer theory which is why we require a primitive ℓth root of unity in the field. In this situation our desired subfield is then given by a defining equation of the form $X^\ell - \alpha$.

We give an explicit algorithm for computing extensions of degree $\ell = 2$. This can easily be generalized to compute extensions of prime degree ℓ of K contained in the Hilbert class field, as long as K contains a primitive ℓth root of unity. A field K always contains -1, the 2nd root of unity, so we can always construct an extension of degree $\ell = 2$ if one exists. An ℓth root can be added but at the expense of increasing the degree of the number field. In general, K will only have an extension of degree ℓ contained in the Hilbert class field if the class group is divisible by ℓ. So Algorithm 4 for $\ell = 2$ requires that the order of the class group be even. Algorithm 4 below uses virtual units and builds on the unit and class group. We must show that we can compute valuations of elements that are only given in the Log representation. We show that this can be done by relating them to the factorization of certain ideals.

We now assume that $\ell = 2$. The group of virtual units (modulo powers of 2), $V_2(K)/(K^*)^2$, is generated by a basis for the unit group and the generators α from ideals $g^d = \alpha\mathcal{O}_K$, where g is a generator of the class group of order d with $2 \mid d$. Since the elements are taken modulo $(K^*)^2$, $\alpha^2 = 1$ for any α, and the number of generators is $r_c + r_u + 1$, where r_u is the rank of the unit group (so $r_u + 1$ accounts for the generator of torsion part of the unit group), and r_c is the 2-rank of the class group.

The following algorithm is a special case of Algorithm 5.2.14 in [Coh00] with trivial modulus $\mathfrak{m}$. We also change the representation of elements to a compact representation for efficiency. It computes a subfield of Hilbert class field rather than subfields of the ray class field. In this case we can analyze the running time of the algorithm.

Algorithm 4. Degree two subextension of the Hilbert class field of K
Input: Number field K whose class group has even order, $\mathcal{O}_K$.
Output: The generator $\mathrm{Log}(\alpha)$ of the generating polynomial $X^2 - \alpha$ of the extension.

1. *Compute generators $v_1 = Log(\varepsilon_1), ..., v_{r_u+1} = Log(\varepsilon_{r_u+1})$ of the unit group, and a basis of the class group $Cl(K) = \bigoplus_{i=1}^{g_c} (\mathbb{Z}/d_i\mathbb{Z})\bar{\mathfrak{a}}_i$ ordered such that the first $d_1, ..., d_{r_c}$ are divisible by 2. For each generator $\mathfrak{a}_i$ of the class group, compute $v_{r_u+1+i} = Log(\alpha_i)$, where $\alpha_i\mathcal{O}_K = \mathfrak{a}_i^{d_i}$.*

2. *Factor each generator $\mathfrak{a}$ of the class group as $\mathfrak{a} = \prod \mathfrak{p}_j^{s_j}$. Then $\mathfrak{a}^d = \prod \mathfrak{p}_j^{ds_j}$.*

3. *Factor* $2\mathcal{O}_K = \Pi_{i=1}^m \mathfrak{p}_i^{e_i}$ *into powers of prime ideals with* $e_i > 0$. *For each* $\mathfrak{p}_i$ *compute an element* $\pi_i \in \mathfrak{p}_i - \mathfrak{p}_i^2$.

4. *For each* α_j *and each* $\mathfrak{p}_i$ *with* $1 \leq j \leq r_c, 1 \leq i \leq m$, *let* $\alpha_{ij} = \alpha_j / \pi_i^{d_j s_i}$, *so* α_{ij} *is coprime to* $\mathfrak{p}_i$, *and compute it as* $Log(\alpha_{ij}) = Log(\alpha_j) - d_j s_i Log(\pi_i)$.

5. *Compute the following matrix* M *with* $(r_u + 1 + r_c)$ *columns. For the first* $r_u + 1$ *columns, column* j *has* m *blocks, where the* i*th block consists of the exponents of the discrete log of* ε_j *in* $(\mathcal{O}_K/\mathfrak{p}_i^{2e_i})^*$ *relative to a set of generators from decomposing that group.*

   For the last r_c *columns, the* i*th block of column* r_{u+1+j} *consists of the exponents of the discrete log of* α_{ij} *in* $(\mathcal{O}_K/\mathfrak{p}_i^{2e_i})^*$ *relative to a set of generators from decomposing that group.*

6. *Reduce the coefficients of* M *modulo 2, pick a nonzero vector* $x \in \mathbb{F}_2^{r_c+r_u+1}$ *in the kernel of* M, *and output* $Log(\alpha) = \sum_i x_i v_i$.

Theorem 5.1. *Let* K *be a number field containing an* ℓ*th root of unity for a fixed prime* ℓ. *Computing a degree* ℓ *extension of* K *contained in the Hilbert class field of* K *reduces to computing the unit group and class group of* K, *solving the principal ideal problem in* K, *factoring ideals, and computing discrete logs in finite groups in polynomial time.*

Proof. For ease of notation the algorithm above is for the case $\ell = 2$. It can be easily modified to compute an α such that $X^\ell - \alpha$ is a degree ℓ extension inside the Hilbert class field.

If $\mathfrak{a}_j$ factors as $\Pi \mathfrak{p}_i^{s_i}$ then $v_{\mathfrak{p}_i}(\mathfrak{a}_j) = s_i$. Hence $v_{\mathfrak{p}_i}(\mathfrak{a}_j^{d_j}) = d_j s_i$. Thus the valuation of α_j at the prime $\mathfrak{p}_i$, $v_{\mathfrak{p}_i}(\alpha_j)$, equals $d_j s_i$. Hence the above algorithm shows that we can compute the valuation of α_j from only the Log representation. Since $\alpha_{ij} = \alpha_j / \pi_i^{d_j s_i}$ is coprime to $\mathfrak{p}_i$ and to powers of $\mathfrak{p}_i$ it reduces to computing discrete logs of α_{ij} in $(\mathcal{O}_K/\mathfrak{p}_i^{2e_i})^*$.

Factoring $2\mathcal{O}_K$ reduces to factoring as shown in Section 4. □

6 Computing the maps for the ray class group

In this section we show how to compute images and preimages for the three maps ρ, ψ, ϕ in the exact sequence

$$U(K) \xrightarrow{\rho} (\mathcal{O}_K/\mathfrak{m})^* \xrightarrow{\psi} \mathrm{Cl}_{\mathfrak{m}}(K) \xrightarrow{\phi} \mathrm{Cl}(K) \to 1, \tag{1}$$

as they are necessary for computing the ray class group $\mathrm{Cl}_{\mathfrak{m}}(K)$. Recall that the ray class group of a number field was defined to be as the set of ideals coprime to $\mathfrak{m}$ modulo the principal ideals coprime to $\mathfrak{m}$ that can be generated by an element α with $\alpha \equiv 1 \bmod {}^*\mathfrak{m}$.

We now define the maps that appear in this exact sequence. There is a natural group homomorphism ρ from the elements of K^* which are coprime to $\mathfrak{m}$ into $(\mathcal{O}_K/\mathfrak{m})^*$ defined as follows. If α is such an element in K^*, then α can be written as β/γ with $\beta, \gamma \in \mathcal{O}_K$ and coprime to $\mathfrak{m}$. We define the class $\overline{\alpha} \in (\mathcal{O}_K/\mathfrak{m}_0)^*$ to be $\overline{\alpha} := \overline{\beta}/\overline{\gamma}$. This does not depend on the choice of β and γ. Then $\rho(\alpha)$ is defined to be $\rho(\alpha) = (\overline{\alpha}, \mathrm{sign}(\sigma_i(\alpha))_{\sigma_i \in \mathfrak{m}_\infty})$. We denote the restriction of this map to $U(K)$ by ρ again, and this is the first map in the exact sequence.

To define the map ψ we proceed as follows. By strong approximation, any element of $(\mathcal{O}_K/\mathfrak{m})^*$ can be written as $\rho(\alpha)$, for some $\alpha \in \mathcal{O}_K$ which is coprime to $\mathfrak{m}$. The map ψ is then defined by sending an element $\rho(\alpha)$ to the ideal class of $\mathrm{Cl}_{\mathfrak{m}}(K)$ represented by $\alpha\mathcal{O}_K$. The map ϕ sends an ideal class of $\mathrm{Cl}_{\mathfrak{m}}(K)$ represented by an ideal $\mathfrak{a}$ to the ideal class of $\mathrm{Cl}(K)$ represented by $\mathfrak{a}$. The map ϕ is surjective by the approximation theorem for rings of integers.

6.1 Image of $\psi : (\mathcal{O}_K/\mathfrak{m})^* \to \mathbf{Cl}_\mathfrak{m}(K)$.

First we show how to map an element $(\bar{\alpha}, v) \in (\mathcal{O}_K/\mathfrak{m})^*$ to an element of $\mathrm{Cl}_\mathfrak{m}(K)$. In order to do this, $(\bar{\alpha}, v)$ must be mapped to an element $\beta \in \mathcal{O}_K$ such that $\alpha \equiv \beta \bmod^* \mathfrak{m}_0$ and such that the signature of β, which is the sign homomorphism from K^* to $\mathbb{F}_2^{|\mathfrak{m}_\infty|}$, equals v, i.e. $s(\beta) = v$. By the definition of ψ, the desired image of $(\bar{\alpha}, v)$ under ψ is then $\beta\mathcal{O}_K$. Cohen [Coh00, page 205] gives a heuristic to do this computation. We show how to reduce this to an approximate closest vector problem.

The goal is to compute elements $(1 + \beta_j)$, $\beta_j \in \mathfrak{m}_0$, such that the sign of $(1 + \beta_j)$ is negative under the jth embedding and positive under the other embeddings. Then to obtain an element β with signature v, we start with α and adjust its signature while staying in the same equivalence class modulo $\mathfrak{m}_0$. That is, we take $\beta = \alpha \cdot \Pi_i(1+\beta_i)$, where i is over the coordinates of $s(\alpha)$ that are different from v. Then β is clearly congruent to α since $(1 + \beta_j) \equiv 1 \bmod \mathfrak{m}_0$, and α was modified by a product of such elements.

To compute $\beta_j \in \mathfrak{m}_0$ such that the jth coordinate of $(1+\beta_j)$ has negative sign and the rest have positive sign we proceed as follows. Let $M > 2^n \det(\mathfrak{m}_0)$. We can find a closest vector to a point in $\mathbb{Q}^k$ to within a factor of 2^k. Let $x = (2^{k+1}M, ..., 2^{k+1}M, -2^{k+1}M, 2^{k+1}M, ..., 2^{k+1}M)$, where the negative coordinate is at position j. Call CVP on the ideal lattice $\mathfrak{m}_0$ and x, and let y be the vector returned [Len92]. Convert y into an element $y' \in \mathfrak{m}_0$, and output $1 + y'$. The conversion can be done by just outputting the first coordinate since we can choose our original embedding such that the first coordinate is the identity. By choice of M, we know there is a lattice point within M of x. Using CVP we get a point within $2^k M$. Therefore, the sign of each coordinate is preserved, even after adding 1 to it. The element β will have polynomial representation size. Each β_i has size bounded $\log(2^n \det(\mathfrak{m}_0) k 2^{k+1})$, so the product of n is small. Therefore the output $\beta\mathcal{O}_K$ can be computed in time $\mathrm{poly}(n, \log \Delta_K, \log \mathcal{N}(\mathfrak{m}_0))$.

6.2 Preimage of ψ and image of ρ.

In this section we modify the approach in [Coh00] by using the multiplicative representation of the units, rather than using the exponentially large standard representation. We must show that we can still get the same output even though we only have access to the compact representations, which are polynomial-size representations of the same object.

Let $\overline{g} \in \mathrm{Cl}_\mathfrak{m}(K)$ be an element in the image of ψ. Such an element is represented by an ideal g of $\mathcal{O}_K$ which is coprime to $\mathfrak{m}_0$. One challenge with this map is that while we know that $g = \alpha\mathcal{O}_K$ for some $\alpha \in K^*$ (since the sequence is exact), we cannot take it modulo $\mathfrak{m}_0$ since α may be not in $\mathcal{O}_K$. It is also not possible to express α as a quotient of elements of $\mathcal{O}_K$ and output these elements since they will in general be too large. Instead, the next algorithm finds $\overline{\beta}\overline{\gamma}^{-1} \in (\mathcal{O}_K/\mathfrak{m}_0)^*$ of polynomial size and a vector $v \in \mathbb{F}_2^{|\mathfrak{m}_\infty|}$ such that $\psi(\overline{\beta}\overline{\gamma}^{-1}, v) = \overline{g}$.

Algorithm 5. Preimage of ψ
Input: Ideal $g \subseteq \mathcal{O}_K$ such that $\overline{g} \in \mathrm{Cl}_\mathfrak{m}(K)$ is in the image of ψ.
Output: Element $\bar{\beta}\bar{\gamma}^{-1}$ and vector v such that $\psi(\overline{\beta}\overline{\gamma}^{-1}, v) = \overline{g}$.

1. *We have $\phi(\overline{g}) = \overline{g}$, which is trivial in $Cl(K)$, because the sequence (1) is exact. Hence $g = \alpha\mathcal{O}_K$, for some $\alpha \in K^*$ and α is coprime to $\mathfrak{m}_0$ because g is.*

   Compute a multiplicative representation of α using the principal ideal algorithm in $Cl(K)$.

2. *Compute d, the lcm of the denominators of the coordinates of α. This is the denominator in input, which is the HNF of the input $g = \alpha\mathcal{O}_K$.*

3. *Compute* $\mathfrak{b} = d\mathcal{O}_K + \mathfrak{m}_0$. *Factor* $\mathfrak{b}$ *as* $\mathfrak{b} = \Pi \mathfrak{p}^{v_\mathfrak{p}}$ *into powers of prime ideals.*

4. *Compute* $k = \max_{\mathfrak{p}|\mathfrak{b}} \lfloor v_p(d)e(\mathfrak{p})/v_\mathfrak{p} \rfloor + 1$. *The quantity* $e(\mathfrak{p})$, *i.e. the power of* $\mathfrak{p}$ *in* $p\mathcal{O}_K$, *was computed when we factored* $p\mathcal{O}_K$. *Here* p *is the prime number such that* $(p) \subseteq \mathfrak{p}$, *and* $v_p(d)$ *denotes the exponent of* p *in* d.

5. *Compute* $\mathfrak{d} = d\mathcal{O}_K + \mathfrak{b}^k$ *and* $\mathfrak{d}^{-1}$. *Compute* $a \in d\mathfrak{d}^{-1}$ *and* $c \in \mathfrak{b}^k\mathfrak{d}^{-1}$ *such that* $a + c = 1$, *by applying Lemma 3.1 part (1) below with* $\mathfrak{a} = d\mathfrak{d}^{-1}, \mathfrak{c} = \mathfrak{b}^k\mathfrak{d}^{-1}$.

6. *Let* $\beta = a\alpha$ *and* $\gamma = a$. *Then* $\beta, \gamma \in \mathcal{O}_K$, *and* $\overline{\beta} \cdot \overline{\gamma}^{-1} \in (\mathcal{O}_K/\mathfrak{m}_0)^*$.

7. *Let* $(\beta_1, ..., \beta_k, e_1, ..., e_k)$ *be the multiplicative representative of* β. *Compute*

$$\beta + \mathfrak{m}_0 = \beta_1^{e_1} \cdots \beta_k^{e_k} + \mathfrak{m}_0$$

   by iterating through each i, *multiplying by the next* β_i, *and reducing mod* $\mathfrak{m}_0$. *This keeps the representation size small. Multiply by* γ^{-1}.

8. *Compute the signature* v *of* $\beta\gamma^{-1}$ *using the multiplicative representation of* β.

9. *Output* $\overline{\beta}\overline{\gamma}^{-1}$ *and the signature* v.

Lemma 6.1. *Computing the preimage of a given element under* ψ *reduces to computing the unit and class group of* K, *the principal ideal problem in* K, *and factoring integers, in time poly*$(2^n, \log \Delta_K)$.

Proof. The algorithm above shows that compact representations can be used to compute the image of ψ.

Factoring $\mathfrak{b}$ is efficient since we have the factorization of $\mathfrak{m}_0$, and $\mathfrak{m}_0 \subseteq \mathfrak{b}$. The only possible primes appearing in the factorization of $\mathfrak{b}$ are those appearing in the factorization of $\mathfrak{m}_0$, and so computing valuations at these primes is sufficient.

Given the multiplicative representation, computing modulo the ideal to keep the elements polynomial size. Computing the signature of β can be done efficiently also by using the multiplicative representation and using the fact that the signature map is a homomorphism $K^* \to \mathbb{F}_2^{|\mathfrak{m}_\infty|}$. □

To see that the image of an element under the map ρ can be computed efficiently we do the following. Given an element $\varepsilon \in U(K)$ in a multiplicative representation and then map it into $(\mathcal{O}_K/\mathfrak{m})^*$ by following Step 7 above and by computing the signature of ε by using the multiplicative representation as explained above.

6.3 Preimage and image of $\phi : \mathbf{Cl}_\mathfrak{m}(K) \to \mathbf{Cl}(K)$.

Elements of the class group are represented by ideals of $\mathcal{O}_K$ while elements of the ray class group are represented by ideals which are coprime to $\mathfrak{m}$. Hence, inverting ϕ requires taking any element in the class group represented by an ideal $\mathfrak{a}$ and computing an equivalent ideal $\mathfrak{b}$ (in the class group) which is coprime to $\mathfrak{m}$. This can be done using the factorization of $\mathfrak{m}$ using Corollary 1.3.9 in [Coh00] in time poly$(n, \log \Delta, \log \mathcal{N}(\mathfrak{m}_0))$.

Evaluating ϕ is trivial, because this is simply the identity map $\overline{g} \mapsto \overline{g}$.

7 Computing ray class groups

In this section we describe how to compute ray class groups and how to solve the principal ideal problem in the ray class group given the algorithms in Section 3 and for the maps in Section 6.

We will follow the strategy described in [CDO98] which computes the ray class group as a group extension using the following four-term right-exact sequence:

$$U(K) \xrightarrow{\rho} (\mathcal{O}_K/\mathfrak{m})^* \xrightarrow{\psi} \mathrm{Cl}_{\mathfrak{m}}(K) \xrightarrow{\phi} \mathrm{Cl}(K) \to 1.$$

Theorem 7.1. *Computing the ray class group of a field K, given a modulus $\mathfrak{m}$ and $\mathcal{O}_K$, reduces to computing the unit and class group of K, discrete logs in the class group and finite groups, the principal ideal problem in K and factoring $\mathfrak{m}_0$, in time poly$(2^n, \log \Delta_K, \log \mathcal{N}(\mathfrak{m}_0))$.*

Proof. Algorithm 5.1 of [CDO01] discusses how to obtain generators and relations for a finite group G that is part of a four-term right-exact sequence

$$A \to B \to G \to C \to 1.$$

Computing the ray class group by this method requires generators for the unit group $U(K)$, generators and relations for the class group $\mathrm{Cl}(K)$, as well as solving the discrete log problem and the principal ideal problem in $\mathrm{Cl}(K)$. We also require generators and relations for the group $(\mathcal{O}_K/\mathfrak{m})^*$ and discrete log computations in this group, the ability to compute the image of an element for ρ, ψ, ϕ, and the ability to compute a preimage of an element for ψ and ϕ.

In Section 3 we showed that computing generators and relations for the finite group $(\mathcal{O}_K/\mathfrak{m})^*$ reduces to factoring $\mathfrak{m}_0$ and discrete log in finite groups (just order finding) in time $\mathrm{poly}(n, \log \Delta_K, \log \mathcal{N}(\mathfrak{m}_0))$. In Section 6 we showed that computing images and preimages of the required maps reduces to factoring $\mathfrak{m}_0$ in time $\mathrm{poly}(2^n, \log \Delta_K)$ □

8 Examples

Below we give several examples of factorizations of ideals in number fields, of ray class groups and Hilbert class fields.

1. Let $K := \mathbb{Q}[\sqrt{m}]$, where m is a square-free integer. Then

   $$\mathcal{O}_K = \{a + b\sqrt{m} : a, b \in \mathbb{Z}\}$$

   if $m \equiv 2$ or $3 \pmod 4$ and

   $$\mathcal{O}_K = \{(a + b\sqrt{m})/2 : a, b \in \mathbb{Z}, a \equiv b \pmod 2\}$$

   if $m \equiv 1 \pmod 4$ [Mar77, p. 15].

   Suppose that p is an odd prime dividing m. Then $\mathfrak{p} := (p, \sqrt{m})$ is a prime ideal of $\mathcal{O}_K$ and $p\mathcal{O}_K = (p, \sqrt{m})^2$. Hence $v_{\mathfrak{p}}(p\mathcal{O}_K) = 2$. The Galois group $\mathrm{Gal}(K/\mathbb{Q})$ has two elements, so it is cyclic of order 2.

2. Let $L := \mathbb{Q}(\zeta_n)$ with ζ_n a primitive nth root of unity (say $\zeta_n = e^{2\pi i/n}$). The ring of integers of L is $\mathcal{O}_L = \mathbb{Z}[\zeta_n]$ [Mar77, p. 35]. We have $[L : \mathbb{Q}] = \phi(n)$, and the map sending an integer k which is coprime to n to the automorphism $\zeta_n \mapsto \zeta_n^k$ gives an isomorphism of $\mathrm{Gal}(L/\mathbb{Q})$ with $(\mathbb{Z}/nZ)^*$ [Mar77, p. 18].

   Now let ζ_p be a pth root of unity with p prime. The ideal $\mathfrak{P}$ generated by $(1 - \zeta_p)$ is a prime ideal of $\mathcal{O}_L = \mathbb{Z}[\zeta_p]$ and $p\mathcal{O}_L = (1 - \zeta_p)^{(p-1)}$ [Mil08a, p. 90]. So $v_{\mathfrak{P}}(p\mathcal{O}_L) = p - 1$.

3. It is not always the case that the ring of integers can be generated by one element over $\mathbb{Z}$. The following example goes back to Dedekind. Let $M := \mathbb{Q}(\alpha)$, where α is a root of $X^3 + X^2 - 2X + 8$. Then $\mathcal{O}_M \neq \mathbb{Z}[\beta]$ for any $\beta \in \mathcal{O}_K$ ([Mil08a, p. 32]). One can show that $2\mathcal{O}_M$ factors as a product of 3 distinct prime ideals which implies that $2 \mid [\mathcal{O}_M : \mathbb{Z}[\beta]]$ for any $\beta \in \mathcal{O}_M$ by Theorem 4.2.

4. Let $K := \mathbb{Q}(\sqrt{-5})$. The Hilbert class field of K has degree 2 over K since the class group of K has two elements [Mar77, p. 133]. It is not hard to check that $\mathbb{Q}[\sqrt{-5}, i]$ is unramified over K at all finite and infinite places, and so by degree considerations it must be the Hilbert class field of K.

5. Let $K := \mathbb{Q}(\sqrt{3})$, and let $\mathfrak{m}$ be the modulus that consists of the unit ideal together with the two real embeddings of K. The number field K has trivial class group, but $\mathrm{Cl}_{\mathfrak{m}}(K)$ has order 2 [Mil08b, p. 148].

6. For the field $\mathbb{Q}$ and the modulus $\mathfrak{m} = (p)\infty$ with p any odd prime, the ray class group is cyclic of order $p - 1$, $\mathrm{Cl}_{\mathfrak{m}} \cong (\mathbb{Z}/p\mathbb{Z})^*$. The ray class group for the modulus (p) is isomorphic to $(\mathbb{Z}/p\mathbb{Z})^*/\{\pm 1\}$ [Mil08b, p. 155].

7. Let $K_1 := \mathbb{Q}(\sqrt{-30030})$. Then K_1 has an infinite Hilbert class field tower [Roq67]. The field $K_2 := \mathbb{Q}(\sqrt{9699690})$ has an infinite class field tower as well [Roq67]. The proof, however, is not constructive.

9 Efficient quantum algorithms in constant degree number fields

The core problems listed in the reductions have efficient quantum algorithms. Assume a constant degree number field is given. Computing the ring of integers reduces to factoring. See [Chi89] and [BL94, Theorem 1.3]. The Log representation of generators of the unit group can be efficiently computed by a quantum algorithm [Hal05, SV05] and then transformed into a multiplicative representation [Thi95]. The class group can be computed and the principal ideal problem can be solved efficiently by a quantum algorithm. Discrete log in finite groups (assuming unique efficiently computable representatives) can be computed in quantum polynomial time [Sho97].

Lemma 9.1. *There is a polynomial-time quantum algorithm that factors fractional ideals of a number field K into a product of prime ideals of $\mathcal{O}_K$.*

Proof. In Section 4 we show that factoring ideals in number fields reduces to factoring integers. □

Corollary 9.2. *There is a polynomial-time quantum algorithm that computes the ray class group of a constant degree number field K, given a modulus $\mathfrak{m}$.*

Proof. There is a polynomial-time quantum algorithm for computing the preimage of a given element under ψ in constant degree number fields: Computing the generator $\mathrm{Log}(\alpha)$ is possible with an efficient quantum algorithm. Mapping the Log representation of α into $(\mathcal{O}_K/\mathfrak{m})^*$ can be done by first converting it into a multiplicative representation [Thi95]. Also, the group $(\mathcal{O}_K/\mathfrak{m})^*$ can be computed in quantum polynomial time. Computing the unit group and class group are efficient in the constant degree case. □

Corollary 9.3 (Consequence of Theorem 5.1)**.** *There is an efficient quantum algorithm to compute degree 2 extensions inside the Hilbert class field of constant degree number fields K.*

Corollary 9.4. *There is an efficient quantum algorithm for computing discrete logs in the ray class group and for the principal ideal problem in the ray class group of a constant degree number field.*

References

[BBS09] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 55–64, New York, NY, USA, 2009. ACM.

[BL94] J. A. Buchmann and H. W. Lenstra, Jr. Approximating rings of integers in number fields. *J. Théor. Nombres Bordeaux*, 6(2):221–260, 1994.

[BS08] J. P. Buhler and P. Stevenhagen, editors. *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*, volume 44 of *Mathematicial Science Research Institute publications*. Cambridge University Press, 2008.

[CDO98] H. Cohen, F. Diaz y Diaz, and M. Olivier. Computing ray class groups, conductors and discriminants. *Math. Comp.*, 67(222):773–795, 1998.

[CDO01] Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier. Algorithmic methods for finitely generated abelian groups. *J. Symbolic Comput.*, 31(1-2):133–147, 2001. Computational algebra and number theory (Milwaukee, WI, 1996).

[Chi89] A. L. Chistov. The complexity of the construction of the ring of integers of a global field. *Dokl. Akad. Nauk SSSR*, 306(5):1063–1067, 1989.

[Coh93] Henri Cohen. *A course in computational algebraic number theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[Coh00] Henri Cohen. *Advanced topics in computational number theory*, volume 193 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.

[Gur03] Venkatesan Guruswami. Constructions of codes from number fields. *IEEE Trans. Inf. Th.*, 49(3):594–603, 2003.

[Hal05] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474, 2005.

[Hun80] Thomas W. Hungerford. *Algebra*, volume 73 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1980. Reprint of the 1974 original.

[Len92] H.W. Lenstra. Algorithms in algebraic number theory. *Bulletin of the AMS*, 26(2):211–244, 1992.

[Mar77] Daniel A. Marcus. *Number fields*. Springer-Verlag, New York, 1977. Universitext.

[Mil08a] James S. Milne. Algebraic number theory (v3.01), 2008. Available at www.jmilne.org/math/.

[Mil08b] J.S. Milne. Class field theory (v4.00), 2008. Available at www.jmilne.org/math/.

[PR07] Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 478–487, New York, NY, USA, 2007. ACM Press.

[Rab80] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM Journal on Computing*, 9(2):273–280, 1980.

[Roq67] Peter Roquette. On class field towers. In *Algebraic Number Theory (Proc. Instructional Conf., Brighton, 1965)*, pages 231–249. Thompson, Washington, D.C., 1967.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Ste08] Peter Stevenhagen. The arithmetic of number rings. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 209–266. Cambridge Univ. Press, Cambridge, 2008.

[SV05] Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005.

[Thi95] Christoph Thiel. *On the complexity of some problems in algorithmic algebraic number theory.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1995.