THE OPEN BOOK SERIES 1

ANTS X
Proceedings of the Tenth
Algorithmic Number Theory Symposium

Computing the unit group, class group, and compact representations in algebraic function fields

Kirsten Eisenträger and Sean Hallgren

msp

THE OPEN BOOK SERIES **1** (2013)
Tenth Algorithmic Number Theory Symposium
dx.doi.org/10.2140/obs.2013.1.335

Computing the unit group, class group, and compact representations in algebraic function fields

Kirsten Eisenträger and Sean Hallgren

Number fields and global function fields have many similar properties. Both have many applications to cryptography and coding theory, and the main computational problems for number fields, such as computing the ring of integers and computing the class group and the unit group, have analogues over function fields. The complexity of the number field problems has been studied extensively, and quantum computation has provided exponential speedups for some of these problems. In this paper we study the analogous problems in function fields. We show that there are efficient quantum algorithms for computing the unit group, for computing the class group, and for solving the principal ideal problem in function fields of arbitrary degree. We show that compact representations exist, which allows us to show that the principal ideal problem is in NP. We are also able to show that these compact representations can be computed efficiently, in contrast with the number field case.

1. Introduction

Algebraic number theory is concerned with the study of *number fields* — that is, finite extensions L of $\mathbb{Q}$ — and of the rings of algebraic integers $\mathbb{O}_L$ of such L. Similarly, we can consider finite algebraic extensions K of $\mathbb{F}_q(t)$, where $\mathbb{F}_q(t)$ is the quotient field of the polynomial ring $\mathbb{F}_q[t]$. These fields are called *function fields over finite fields* or *global function fields*. It was noticed early on that the integers have many properties in common with $\mathbb{F}_q[t]$, and similarly, that number fields and global function fields have many similar properties. Often, a problem that is posed for number fields admits an analogous problem for global function fields, and the other way around. For example, the Riemann hypothesis for the classical Riemann

MSC2010: primary 11Y16; secondary 11R27, 11R29.
Keywords: function fields, compact representations, infrastructure, unit group, principal ideal problem.

zeta function $\zeta(s)$ is still open, while the function-field analogue of this conjecture was proved by Weil.

The main computational problems for number fields include computing the ring of integers, the class group, and the unit group, and solving the principal ideal problem. These problems have been studied extensively, and there are a large number of classical algorithms for computing with number fields. Applications include the number field sieve, which is the fastest classical algorithm for factoring [29], and the Buchmann-Williams key-exchange system, whose security depends on the hardness of the principal ideal problem [7]. The recent push to make lattice-based cryptography more efficient has relied upon special lattices that come from number fields [33; 30]. Error-correcting codes have also been constructed using such lattices [23]. Quantum algorithms have been the source of exponential speedups for many of these computational problems for number fields. There are efficient quantum algorithms for computing the unit group and class group, and for solving the principal ideal problem in constant degree number fields [25; 38]. Some field extensions have also been computed using quantum algorithms [16]. In this paper we study the analogous computational problems over function fields.

Function fields also have many applications in cryptography and coding theory. There are many cryptographic applications that use elliptic curves or Jacobians of curves of small genus defined over finite fields [13]. Most of these rely on the assumption that the discrete log problem is difficult to solve in the underlying group associated with these curves. Another way to state this is that the discrete log problem is assumed to be hard in the divisor class group of the function field of the curve. Error correcting codes have also been based on function fields [22]. In a recent paper, Guruswami [24] constructed codes where everything was efficient except computing the basis for the Riemann-Roch space of a certain divisor.

For number fields the problems listed above have been studied extensively, and they appear to be computationally hard. For example, computing the ring of integers requires squarefree factorization of integers. The best known classical algorithms for computing the unit group, for computing the class group, and for solving the principal ideal problem are exponentially slower than factoring. On the other hand, computing the class group and unit group is in NP$\cap$coNP for arbitrary degree number fields [42], while the quantum algorithms are only efficient for constant degree number fields. One apparent obstacle is that the only way known to compute with ideals of number fields requires a shortest vector problem in ideal lattices to be solved during computations, in order to keep representation sizes small.

In this paper we examine these computational problems over function fields of arbitrary degree. For function fields, computing the ring of integers is computationally equivalent to factoring polynomials over a finite field, which can be done in (classical) polynomial time, so one might hope that much more can be done.

In fact, even the analogue of the shortest vector problem has an efficient classical algorithm. But problems such as computing the divisor class group should be hard classically since they include as a special case the discrete log problem on an elliptic curve (a curve of genus one whose function field has degree two). For certain special classes of function fields (where the degree is two and the genus is large) there are subexponential algorithms for computing the class group, which make them less secure for cryptographic purposes: In [4] the authors give a subexponential algorithm for computing the class group of a hyperelliptic curve of large genus, and [31] gives a subexponential probabilistic algorithm for computing the class group of a real quadratic congruence function field of large genus. In [37] it is shown that various decision problems for quadratic congruence function fields of large genus are in $\mathrm{NP} \cap \mathrm{coNP}$. There are also some exponential algorithms known for more general function fields. Another important computational problem that only exists in the function field case is that of computing Riemann-Roch spaces.

In this paper we show that the principal ideal problem over function fields of arbitrary degree is in NP. To do this we show that compact multiplicative representations exist for elements in function fields. This answers a question of Smart [40] and generalizes [36], which showed the existence of compact representations for real quadratic congruence function fields (which have degree two). Our work adapts work of Thiel, who used compact representations in number fields and showed that the principal ideal problem, the computation of class numbers, and the computation of compact representations of units are in $\mathrm{NP} \cap \mathrm{coNP}$ for number fields [42]. We also show that, unlike the situation for number fields, compact representations can be computed in (classical) polynomial time for arbitrary degree function fields. The standard representation of an element, for example a unit, may take exponentially many bits to represent. Compact representations give a certain factored form of the element which only requires polynomial representation size.

Given this setup, we also show that there are efficient quantum algorithms for computing the unit group and the class group, and for solving the principal ideal problem in arbitrary degree function fields. This is in contrast to the number field case, where currently only the constant degree case has quantum algorithms. These problems are solved by setting up abelian hidden subgroup problems.

One open question related to our work is whether the function field analogues of the problems treated by Thiel are also in $\mathrm{NP} \cap \mathrm{coNP}$. Compact representations played a key role in the number field case. One issue in the function field case is that it is not known how to deterministically compute generators for the class group efficiently.

Another open question is finding an efficient quantum algorithm for computing class field towers of function fields. Certain towers of function fields — namely, Hilbert class field towers — have applications to coding theory. When the tower is infinite one can construct asymptotically good sequences of codes from the fields in such towers [20, p. 212]. Infinite towers are known to exist [39], but for applications of such codes in practice, an explicit construction of the fields in the tower is required. Class groups of certain subrings of the function fields in the tower appear as the Galois groups of the field extensions in the tower. Therefore, computing the class groups (and compact representations), as we do in this paper, is required to compute such towers, as it is in the number field case [16].

In order to set up our algorithms we need efficient algorithms for doing computations in the infrastructure of a function field. Fontein recently provided these and we prove that his algorithms in [19; 17] are polynomial-time. To compute with the infrastructure it is necessary to efficiently compute the Riemann-Roch space of a divisor D. For this we use Hess's algorithm [26], which is a relatively simple, self-contained algorithm. In the appendix we include a complexity analysis of his algorithm. For other references that analyze Hess's algorithm see [19] (which makes some additional assumptions) and [14]. The algorithms above have been implemented, for example in Magma. The focus of this paper, however, is on the complexity analysis. Analyzing the Riemann-Roch algorithm addresses the missing piece for the codes in [24] to be efficient.

One technical challenge in our work is adapting Thiel's algorithm for computing compact representations [42] from the number field case to the function field case. To do this, and to end up with a polynomial-time algorithm, we must show that we can compute compact representations without searching for minima in a region of exponential size, something that is necessary in the number field case. We also analyze the Riemann-Roch space computation. This involves showing that we can efficiently compute certain prime ideals of the ring $\mathcal{O}_\infty$ (see Section 2 for notation) that we use to compute the Riemann-Roch space $L(D)$ from a given representation of the divisor D. We carry out this computation by factoring and computing radicals of certain ideals; further details, and the complexity analysis, can be found in Appendix B. Our algorithm generalizes the ideal factorization algorithm for number fields [16].

There have been other approaches to the study of some of these problems over function fields. In [27] Huang and Ierardi gave a construction of the Riemann-Roch space that is polynomial-time, assuming that all the singular points of the plane curve defining the function field are ordinary and defined over the base field. For another construction, which uses the Brill-Noether method, see Volcheck [43]. Recently, the authors learned that the (unpublished) Habilitation thesis by Diem [14] also studied Hess's algorithm. Kedlaya [28] showed how to compute zeta functions of curves with a quantum algorithm. His method requires computing the size of the divisor class group $\mathrm{Pic}^0(K)$, and he showed how to compute in the group efficiently.

Our work, by contrast, requires the different representation using the infrastructure of Fontein [17] in order to compute in the unit group and the class group, rather than only in the divisor class group $\mathrm{Pic}^0(K)$. The infrastructure also allows us to show the existence of compact representations.

Infrastructures have also been studied in [35; 18], which give quantum algorithms for computing one-dimensional infrastructures and the period lattice of infrastructures of fixed dimension.

2. Background on algebraic function fields and divisors

Algebraic function fields over finite fields. Let k be a finite field with $q = p^m$ elements for some prime p and integer $m > 0$. An *algebraic function field K/k* is an extension field $K \supseteq k$ such that K is a finite algebraic extension of $k(x)$ for some $x \in K$ which is transcendental over k. After replacing k with a finite extension, if necessary, we may assume that k is the *constant field* of K, that is, that k is algebraically closed in K. By [41, p. 144] such an algebraic function field is separably generated; that is, there exist $x, y \in K$ such that $K = k(x, y)$. The function field K is then specified by the finite field k, the indeterminate x and the minimal polynomial $f \in k(x)[T]$ of y over $k(x)$. Throughout the paper, we assume that K is given to us as $K = k(x, y)$ with x, y as above, and we let $d := [K : k(x)]$.

A *valuation ring* of the function field K/k is a ring $\widetilde{\mathcal{O}} \subseteq K$ such that $k \subsetneqq \widetilde{\mathcal{O}} \subsetneqq K$ and such that for every $z \in K$ we have $z \in \widetilde{\mathcal{O}}$ or $z^{-1} \in \widetilde{\mathcal{O}}$. A valuation ring is a local ring; that is, it has a unique maximal ideal [41, p. 2]. A *place* of a function field K/k is defined to be the maximal ideal of some valuation ring of K/k. To each place $\mathfrak{p}$ of K, there is an *associated discrete valuation* $v_{\mathfrak{p}} \colon K^* \to \mathbb{Z}$, and there is a one-to-one correspondence between places of K/k and discrete valuations of K/k [41, pp. 5–6]. Denote by $\mathcal{P}_K$ the set of all places of K. If $\mathfrak{p}$ is a place of K with corresponding valuation ring $\mathcal{O}_{\mathfrak{p}}$, we define the *degree of* $\mathfrak{p}$ to be the degree of the field extension of $\mathcal{O}_{\mathfrak{p}}/\mathfrak{p}$ over k; that is, $\deg \mathfrak{p} = [\mathcal{O}_{\mathfrak{p}}/\mathfrak{p} : k]$. If F/K is an extension of algebraic function fields we say that a place $\mathfrak{P} \in \mathcal{P}_F$ *lies above* a place $\mathfrak{p} \in \mathcal{P}_K$ if $\mathfrak{p} \subseteq \mathfrak{P}$.

For the rational function field $k(x)$, the places are completely understood: The places of $k(x)$ correspond to the irreducible polynomials of $k[x]$, together with a "place at infinity", denoted ∞.

Let v_∞ be the discrete valuation corresponding to the infinite place ∞ of the rational function field $k(x)$. Then v_∞ is defined via $v_\infty(f/g) = \deg g - \deg f$, for $f, g \in k[x]$. Let $\mathfrak{o}_\infty := \{a \in k(x) : v_\infty(a) \geq 0\}$. Then $\mathfrak{o}_\infty$ is the valuation ring associated to v_∞ and the unique maximal ideal of $\mathfrak{o}_\infty$ is generated by $1/x$. Let S denote the set of places of K above ∞. Let

$$\mathcal{O}_\infty := \{a \in K : v_{\mathfrak{p}}(a) \geq 0 \text{ for all } \mathfrak{p} \in S\}.$$

Then $\mathbb{O}_\infty$ is the integral closure of $\mathfrak{o}_\infty$ in K, and $\mathbb{O}_\infty$ is a free $\mathfrak{o}_\infty$-module of rank d. The ring $\mathbb{O}_\infty$ is a principal ideal domain whose prime ideals correspond to the elements in S.

Divisors on algebraic function fields. A *divisor* on K is a formal sum

$$D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p}\,\mathfrak{p}$$

such that $n_\mathfrak{p} = 0$ for all but finitely many $\mathfrak{p}$. Let $\operatorname{Div}(K)$ denote the group of divisors on K. For a divisor D which is given as $D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p}\,\mathfrak{p}$, we define the *degree* of D to be $\deg D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p} \deg \mathfrak{p}$. The divisors of degree zero form a subgroup of $\operatorname{Div}(K)$, which we denote by $\operatorname{Div}^0(K)$. For $f \in K^*$, the *divisor* of f is defined to be

$$\operatorname{div}(f) = \sum_{\mathfrak{p}\in\mathcal{P}_K} v_\mathfrak{p}(f)\,\mathfrak{p}.$$

The set of all divisors of the form $\operatorname{div}(f)$ forms the group $\operatorname{Prin}(K)$ of *principal divisors* on K. Note that if D is a principal divisor then $\deg D = 0$. We define the *divisor class group* $\operatorname{Pic}^0(K)$ to be the quotient of the group of divisors of degree zero by the group of principal divisors; that is, $\operatorname{Pic}^0(K) = \operatorname{Div}^0(K)/\operatorname{Prin}(K)$. The divisor class group is a finite group.

A divisor $D = \sum_\mathfrak{p} n_\mathfrak{p}\,\mathfrak{p}$ is *effective* if $n_\mathfrak{p} \geq 0$ for all $\mathfrak{p}$; we write $D_1 \geq D_2$ to mean that $D_1 - D_2$ is effective. Every divisor D can be written uniquely as $D = D_+ - D_-$ with D^+, D^- effective divisors with disjoint support. We define the *height* of a divisor D to be $\operatorname{ht}(D) := \max\{\deg(D_+), \deg(D_-)\}$. For a divisor $D \in \operatorname{Div}(K)$ we define the *Riemann-Roch space* of D to be the set

$$L(D) := \{f \in K : \operatorname{div}(f) + D \geq 0\} \cup \{0\}.$$

The set $L(D)$ is a vector space over k, and we denote its dimension by $\ell(D)$.

Fractional ideals. Let $\mathbb{O}$ be the integral closure of $k[x]$ in K. Then $\mathbb{O}$ is a free $k[x]$-module of rank d. By [11, Theorem 1], a $k[x]$-basis for $\mathbb{O}$ can be computed in time polynomial in d and $\log q$. If $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$ is the set of places above the infinite place ∞ of $k(x)$, then we also have

$$\mathbb{O} = \{a \in K : v_\mathfrak{p}(a) \geq 0 \text{ for all } \mathfrak{p} \notin S\}.$$

Note that for any nonempty finite set S of places of K one can find an $x \in K$ such that S is the set of infinite places above x. Throughout the paper we assume that $\deg \mathfrak{p}_{n+1} = 1$. This can always be achieved by passing to a finite extension of the constant field k.

A *fractional ideal* of $\mathbb{O}$ is a finitely generated $\mathbb{O}$-submodule of K. Since $\mathbb{O}$ is a Dedekind domain, the nonzero fractional ideals $\operatorname{Id}(\mathbb{O})$ of $\mathbb{O}$ form a (free) abelian group under multiplication. There is a natural homomorphism $\phi\colon \operatorname{Div}(K) \to \operatorname{Id}(\mathbb{O})$ defined by

$$\sum n_{\mathfrak{p}}\mathfrak{p} \mapsto \prod_{\mathfrak{p}\notin S}(\mathfrak{p}\cap\mathbb{O})^{-n_{\mathfrak{p}}}.$$

This map has a right inverse, namely the map $\operatorname{div}\colon \operatorname{Id}(\mathbb{O}) \to \operatorname{Div}(K)$ that sends a fractional ideal $B = \prod_{\mathfrak{p}\notin S}(\mathfrak{p}\cap\mathbb{O})^{n_{\mathfrak{p}}}$ to $\operatorname{div}(B) := -\sum_{\mathfrak{p}\notin S} n_{\mathfrak{p}}\mathfrak{p}$. Hence each divisor can be represented by a pair $\left(A, \sum t_i\mathfrak{p}_i\right)$, where A is a fractional ideal of $\mathbb{O}$ and $\{\mathfrak{p}_1, \dots, \mathfrak{p}_{n+1}\}$ are the places in S, that is, the primes above ∞. This is how we will represent divisors throughout the paper.

The *class group* $\operatorname{Cl}(\mathbb{O})$ of $\mathbb{O}$ is defined to be the group of fractional ideals of $\mathbb{O}$ modulo the principal fractional ideals of $\mathbb{O}$. The class group is a finite abelian group, and the map $\phi\colon \operatorname{Div}(K) \to \operatorname{Id}(\mathbb{O})$ extends to a homomorphism

$$\begin{tikzcd}
\phi\colon \operatorname{Pic}^0(K) \arrow[r] & \operatorname{Cl}(\mathbb{O}) \\
\left[\sum n_{\mathfrak{p}}\mathfrak{p}\right] \arrow[r, mapsto] & \left[\prod_{\mathfrak{p}\notin S}(\mathfrak{p}\cap\mathbb{O})^{-n_{\mathfrak{p}}}\right].
\end{tikzcd}$$

When $\deg \mathfrak{p}_{n+1} = 1$ this map fits into an exact sequence

$$\begin{tikzcd}
0 \arrow[r] & \operatorname{Ker} \arrow[r] & \operatorname{Pic}^0(K) \arrow[r, "\phi"] & \operatorname{Cl}(\mathbb{O}) \arrow[r] & 1.
\end{tikzcd}$$

Here Ker is the subgroup of $\operatorname{Pic}^0(K)$ that is generated by all degree-zero divisors with support in S, so the map $\operatorname{Ker} \to \operatorname{Pic}^0(K)$ is just the inclusion map. Since k is a finite field, Ker is finite by [34, Proposition 14.2, p. 243].

3. Computing efficiently in the unit group

In this section we show how to efficiently compute classically in the unit group of $\mathbb{O}$. Recall that $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_{n+1}\}$ are the places of K above ∞ and that

$$\mathbb{O} = \{a \in K : v_{\mathfrak{p}}(a) \geq 0 \text{ for all } \mathfrak{p} \notin S\}.$$

Also, we assume that $\mathfrak{p}_{n+1}$ is a place of degree 1.

To compute in the unit group, consider the map $\operatorname{val}_\infty : K^* \to \mathbb{Z}^n$ given by $\operatorname{val}_\infty(a) = (-v_{\mathfrak{p}_1}(a), \dots, -v_{\mathfrak{p}_n}(a))$. The image of $\mathbb{O}^*$ under $\operatorname{val}_\infty$ is a lattice Λ in $\mathbb{Z}^n$. By an analogue of Dirichlet's Unit Theorem for function fields, the *unit rank* — that is, the rank of Λ — is equal to $n = \#S - 1$. Since units can have exponentially many bits in the standard representation, computing the unit group means to compute a basis of that lattice, or to compute compact representations for a fundamental set of units as in Definition 4.3. In Lemma 4.7 we show that the compact representation of an element can be computed from its valuation vector, so it follows that these two problems are polynomial time equivalent in function fields.

Fontein [17] showed that it is possible to compute in a finite abelian group which he denotes $\mathrm{Rep}^{f*}(\mathbb{O})$ and which is isomorphic to $\mathbb{Z}^n/\Lambda$. We discuss his approach in the next section. We then show that these computations are efficient. From the group structure of $\mathbb{Z}^n/\Lambda$ we can obtain the basis for the lattice Λ.

3A. ***Minima and reduced ideals in function fields.*** We now give the definitions of minima and reduced ideals and define $\mathrm{Rep}^{f*}(\mathbb{O})$ (see [17]). In the following, by an ideal of $\mathbb{O}$ we will always mean a *fractional* ideal of $\mathbb{O}$.

For each place $\mathfrak{p}_i \in S$, with its associated discrete valuation $v_{\mathfrak{p}_i}$, there is a corresponding *absolute value* $|\alpha|_i$, defined by

$$|\alpha|_i := q^{-v_{\mathfrak{p}_i}(\alpha)\deg \mathfrak{p}_i}.$$

For an ideal A and integers $t_1, \dots, t_{n+1} \in \mathbb{Z}$ we define

$$B(A, (t_1, \dots, t_{n+1})) := \{\alpha \in A : |\alpha|_i \le q^{t_i \deg \mathfrak{p}_i} \text{ for } i = 1, \dots, n+1\}.$$

This is a Riemann-Roch space; we have

$$B(A, (t_1, \dots, t_{n+1})) = L\Big(\mathrm{div}(A) + \sum_{i=1}^{n+1} t_i \mathfrak{p}_i\Big).$$

For an ideal A and $\alpha \in K^*$, let $B(A, \alpha) := B(A, (-v_{\mathfrak{p}_1}(\alpha), \dots, -v_{\mathfrak{p}_{n+1}}(\alpha)))$.

Definition 3.1 (Minima and reduced ideals).

(1) Let A be an ideal of $\mathbb{O}$ and let μ be a nonzero element of A. The element μ is a *minimum* of A if for every nonzero $\alpha \in B(A, \mu)$ we have $|\alpha|_i = |\mu|_i$ for $i = 1, \dots, n+1$.

(2) An ideal A is *reduced* if 1 is a minimum of A.

Denote by $\mathrm{Red}(A)$ the set of reduced ideals of $\mathbb{O}$ which are in the same ideal class as A in $\mathrm{Cl}(\mathbb{O})$. There is a close connection between the set of minima of an ideal A and the set of reduced ideals equivalent to A. First, if μ is a minimum of A and $\epsilon \in \mathbb{O}^*$, then $\epsilon\mu$ is also a minimum of A. This action of $\mathbb{O}^*$ on the set of minima gives rise to a bijection

$$\begin{aligned} \{\text{minima of } A\}/\mathbb{O}^* &\longrightarrow \mathrm{Red}(A) \\ \mu\mathbb{O}^* &\longmapsto (1/\mu)A. \end{aligned}$$

So every element of $\mathrm{Red}(A)$ is of the form $(1/\mu)A$ with μ a minimum of A. Next define a map from the set of reduced ideals equivalent to A to $\mathbb{Z}^n/\Lambda$ by defining

$$\begin{aligned} d\colon \mathrm{Red}(A) &\longrightarrow \mathbb{Z}^n/\Lambda \\ (1/\mu)A &\longmapsto \mathrm{val}_\infty(\mu) + \Lambda \end{aligned}$$

This map is well-defined since $\deg \mathfrak{p}_{n+1} = 1$ (see [17, Corollary 5.3]), and it is also injective [17, Proposition 5.5]. Now we can define Fontein's group $\operatorname{Rep}^{f*}(\mathbb{O})$, which is isomorphic to $\mathbb{Z}^n/\Lambda$.

Definition 3.2. Let A be an ideal of $\mathbb{O}$. An *$f*$-representation* is a tuple

$$(I, (t_1, \dots, t_n)) \in \operatorname{Red}(A) \times \mathbb{Z}^n$$

such that $B(I, (t_1, \dots, t_n, 0)) = k$. Denote the set of all $f*$-representations in $\operatorname{Red}(A) \times \mathbb{Z}^n$ by $\operatorname{Rep}^{f*}(A)$.

When A and B are two ideals that are in the same ideal class in $\operatorname{Cl}(\mathbb{O})$, then clearly $\operatorname{Rep}^{f*}(A) = \operatorname{Rep}^{f*}(B)$. Let

$$\Phi_A\colon \operatorname{Rep}^{f*}(A) \to \mathbb{Z}^n/\Lambda$$

be defined by

$$\Phi_A\left((1/\mu)A, t\right) = \operatorname{val}_\infty(\mu) + t + \Lambda.$$

Here $t = (t_1, \dots, t_n) \in \mathbb{Z}^n$. In [17, Theorem 6.8] it is proved that this map is a bijection. In particular, $\operatorname{Rep}^{f*}(\mathbb{O})$ is isomorphic to $\mathbb{Z}^n/\Lambda$. So to each element (I, t) of $\operatorname{Rep}^{f*}(A)$, there is an associated point in $\mathbb{Z}^n/\Lambda$, and if $I = (1/\mu)A$, we say that (I, t) *represents* the element $\operatorname{val}_\infty(\mu) + t + \Lambda$ of $\mathbb{Z}^n/\Lambda$. Let $[A]$ be the set of ideals equivalent to A in the class group. It is possible to extend Φ_A to a well-defined (but no longer injective) map $\Phi_A\colon [A] \times \mathbb{Z}^n \to \mathbb{Z}^n/\Lambda$ by letting $\Phi_A\left((1/\alpha)A, f\right) = \operatorname{val}_\infty(\alpha) + f + \Lambda$.

In [17, Proposition 8.1] the following is shown:

Proposition 3.3. *Let $(A, (t_1, \dots, t_n))$ be an element of $\operatorname{Rep}^{f*}(B)$ for some ideal B. Then $\operatorname{div}(A) \geq 0$ and $t_i \geq 0$ for $1 \leq i \leq n$. Moreover,*

$$0 \leq \deg \operatorname{div}(A) + \sum_{i=1}^{n} t_i \deg \mathfrak{p}_i \leq g.$$

Here g denotes the genus of the function field.

We want to compute a basis for the n-dimensional lattice Λ. Since $\mathbb{Z}^n/\Lambda$ is isomorphic to $\operatorname{Rep}^{f*}(\mathbb{O})$, it is enough to obtain generators and relations for the finite group $\operatorname{Rep}^{f*}(\mathbb{O})$.

3B. ***Reduction and obtaining generators for* $\operatorname{Rep}^{f*}(\mathbb{O})$.** Let

$$\Phi := \Phi_{\mathbb{O}}\colon \operatorname{Rep}^{f*}(\mathbb{O}) \to \mathbb{Z}^n/\Lambda$$

and its extension to $[\mathbb{O}] \times \mathbb{Z}^n \to \mathbb{Z}^n/\Lambda$ be the maps defined above. The group $\mathbb{Z}^n/\Lambda$ is generated by the standard basis vectors e_i $(1 \leq i \leq n)$, so in order to find generators for $\mathrm{Rep}^{f*}(\mathbb{O})$ we need to find elements $((1/\mu_i)\mathbb{O}, f_i)$ such that

$$\Phi((1/\mu_i)\mathbb{O}, f_i) = e_i + \Lambda.$$

To obtain such elements we consider the elements $(\mathbb{O}, e_i)$, for $i = 1, \dots, n$. These elements are not in $\mathrm{Rep}^{f*}(\mathbb{O})$, but they do have the property that $\Phi(\mathbb{O}, e_i) = e_i + \Lambda$. So to obtain the right elements in $\mathrm{Rep}^{f*}(\mathbb{O})$ we reduce the elements $(\mathbb{O}, e_i)$ to elements $((1/\mu)\mathbb{O}, f_i) \in \mathrm{Rep}^{f*}(\mathbb{O})$ with Algorithm 3.4 below, and use the fact that under Φ, the element $(\mathbb{O}, e_i)$ and its reduction have the same image (see Remark 3.6 below).

The general reduction algorithm that we are describing next works for $\mathrm{Rep}^{f*}(I)$ for any ideal I of $\mathbb{O}$.

Algorithm 3.4 (Reduce).

Input: An ideal A and a vector $t = (t_1, \dots, t_n) \in \mathbb{Z}^n$.

Output: A minimum μ of A, the reduced ideal $(1/\mu)A$, and a vector $t - \mathrm{val}_\infty(\mu)$ such that $((1/\mu)A, t - \mathrm{val}_\infty(\mu)) \in \mathrm{Rep}^{f*}(A)$.

1. Find the minimum ℓ in the interval

$$\Big[-\deg\mathrm{div}(A) - \sum_{i=1}^{n} t_i \deg\mathfrak{p}_i,\ \ g - \deg\mathrm{div}(A) - \sum_{i=1}^{n} t_i \deg\mathfrak{p}_i\Big]$$

   such that $\dim B(A, (t_1, \dots, t_n, \ell)) > 0$.

2. Set $u_1, \dots, u_n = 0$. For each $1 \le i \le n$, increase u_i to find the largest value u_i with $\dim B(A, (t_1 - u_1, \dots, t_n - u_n, \ell)) > 0$.

3. Let μ be a nonzero element of $B(A, (t_1 - u_1, \dots, t_n - u_n, \ell))$. Output $(\mu, (1/\mu)A, (u_1, \dots, u_n))$.

Proposition 3.5. *Algorithm 3.4 is correct and returns* $(\mu, (1/\mu)A, (u_1, \dots, u_n))$ *in time polynomial in* d, $\log q$, $\mathrm{ht}(\mathrm{div}(A))$ *and* $\|t\|_\infty$.

Proof. Let ℓ be minimal such that $\dim B(A, (t_1, \dots, t_n, \ell)) > 0$. By [19, Theorem 4.4.3], we have

$$\ell \in \Big[-\deg\mathrm{div}(A) - \sum_{i=1}^{n} t_i \deg\mathfrak{p}_i,\ \ g - \deg\mathrm{div}(A) - \sum_{i=1}^{n} t_i \deg\mathfrak{p}_i\Big],$$

so the first step of the algorithm requires at most g Riemann-Roch computations. By Theorem B.9, each of these computations

$$B(A, (t_1, \dots, t_n, \ell)) = L\Big(\mathrm{div}(A) + \sum_{i=1}^{n} t_i \cdot \mathfrak{p}_i + \ell \cdot \mathfrak{p}_{n+1}\Big)$$

can be performed in time polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div}(A))$, and $\|t\|_\infty$, because ℓ is at most a polynomial in g, $\mathrm{div}(A)$, and $\|t\|_\infty$, and g is a polynomial in d.

The second step computes the valuation that μ has in the third step. For coordinate i, there are at most t_i Riemann-Roch computations, so in total there are at most $n \max |t_i|$, which is polynomial in d and $\|t\|_\infty$ since $n \le d$. The correctness of steps 2 and 3 follows from the correctness proof of Algorithm 5.4.2 in [19]. □

Remark 3.6. Let A be an ideal of $\mathbb{O}$ and let $t=(t_1,\dots,t_n)\in\mathbb{Z}^n$. Then $(A,(t_1,\dots,t_n))$ represents the same point in $\mathbb{Z}^n/\Lambda$ as its reduction

$$((1/\mu)A, t-\mathrm{val}_\infty(\mu)) \in \mathrm{Rep}^{f*}(A),$$

because

$$\begin{aligned}\Phi_A(A,t) &= t+\Lambda \\ &= \mathrm{val}_\infty(\mu)+(t-\mathrm{val}_\infty(\mu))+\Lambda \\ &= \Phi_A((1/\mu)A, t-\mathrm{val}_\infty(\mu)).\end{aligned}$$

Denote by $\mathrm{Reduce}(A,e)$ the element of $\mathrm{Rep}^{f*}(A)$ computed by Algorithm 3.4. By the above discussion we have $\Phi_A(\mathrm{Reduce}(A,e)) = e+\Lambda$, and if $e' = e+v$ with $v\in\Lambda$, then $\Phi_A(\mathrm{Reduce}(A,e')) = e'+\Lambda = e+\Lambda$. Since $\Phi_A\colon \mathrm{Rep}^{f*}(A)\to\mathbb{Z}^n/\Lambda$ is injective this implies that $\mathrm{Reduce}(A,e)=\mathrm{Reduce}(A,e')$ whenever $e-e'\in\Lambda$.

Definition 3.7. When $\alpha\in K$, the norm of α can be expressed uniquely as $N(\alpha) = f/h$, where f and h are coprime elements of $k[x]$ and h is monic. We define $\mathrm{dg}(N(\alpha))$ to be $\mathrm{dg}(N(\alpha)) = \max\{\deg f, \deg h\}$.

Remark 3.8. When $A=\alpha\mathbb{O}$ then being polynomial in $\mathrm{ht}(\mathrm{div}(A))$ is the same as being polynomial in $\mathrm{dg}\, N(\alpha)$ (see [17, p. 28]).

3C. *Composition and computing inverses in* $\mathbf{Rep}^{f*}(\mathbb{O})$ *and bounding the representation size of elements.*

By [17, Proposition 8.2], elements in $\mathrm{Rep}^{f*}(\mathbb{O})$ can be represented by $O(d^2 g \log q)$ bits. Here g denotes the genus of the function field, which is of size polynomial in d.

Composition of two elements (A, f), (A', f') of $\mathrm{Rep}^{f*}(\mathbb{O})$ is done by multiplying the ideals, adding the two vectors, and then applying Algorithm 3.4 to $(AA', f+f')$. To compute the inverse of $(A, f_1,\dots, f_n)$, compute the inverse A^{-1}, find ℓ minimal such that $\dim B(A^{-1}, (-f_1,\dots,-f_n,\ell)) > 0$ and then reduce using Algorithm 3.4 [19, Proposition 4.3.4]. The ideal arithmetic in $\mathbb{O}$ is polynomial in $\log q$, d, and $\mathrm{ht}(\mathrm{div}(A)), \mathrm{ht}(\mathrm{div}(A'))$ [14, Proposition 2.66, and Proposition 2.69(b)] and $\mathrm{ht}(\mathrm{div}(A))$ is of size polynomial in d and $\log q$ when $(A, f)\in\mathrm{Rep}^{f*}(\mathbb{O})$. Hence Proposition 3.5 implies that composition of two elements and computing inverses in $\mathrm{Rep}^{f*}(\mathbb{O})$ are both polynomial in $\log q$ and d.

4. Compact representations in global function fields

In this section we show how to efficiently compute compact representations of elements $\alpha \in K$ classically. This allows us to show that the principal ideal problem is in NP, and to compute compact representations of units. We adapt the definitions and approach for number fields given in [42, p. 82] to the function field case. The sizes are adapted to match the parameters that are appropriate for number fields and that come from our algorithms. In the function field case we show that an exponential search for minima is no longer necessary.

Definition 4.1. For $\alpha \in K$ and $s \in \mathbb{Q}^n$ we say that α *is close to* s if

$$\|\mathrm{val}_\infty(\alpha) - s\|_1 \leq n + g,$$

where g is the genus of K.

Definition 4.2. A *multiplicative representation* of an element $\alpha \in K$ is a pair

$$M = ((\beta_1, \dots, \beta_\ell), (e_1, \dots, e_\ell)),$$

where $\beta_i \in K$, $e_i \in \mathbb{Z}$, $\ell \in \mathbb{N}$, and such that $\alpha = \prod_{i=1}^{\ell} \beta_i^{e_i}$.

A *binary multiplicative representation* (*BMR*) of an element $\alpha \in K$ is a multiplicative representation such that for all $i \leq \ell$ we have both that $e_i = 2^{\ell - i}$ and that $((\beta_1, \dots, \beta_i), (e_1, \dots, e_i))$ is a minimum of $\mathbb{O}$. Since the exponents e_i are determined, a BMR can be represented as $(\beta_1, \dots, \beta_k)$.

Definition 4.3. A *compact representation* of $\alpha \in K$ is a pair $B = (\gamma, (\beta_1, \dots, \beta_\ell))$, where $(\beta_1, \dots, \beta_\ell)$ is a BMR for a minimum β of $\mathbb{O}$ with $\gamma = \alpha\beta$, and where

$$\begin{aligned} \ell &\leq \log(\|\mathrm{val}_\infty(\alpha)\|_\infty + g), \\ \mathrm{size}(\gamma) &\leq \mathrm{poly}(\log q, d, \mathrm{dg}\, N(\alpha)), \text{ and} \\ \mathrm{size}(\beta_i) &\leq \mathrm{poly}(\log q, d). \end{aligned}$$

Here size denotes the number of bits required to represent the element.

Remark 4.4. Definition 4.3 depends on certain implied constants hidden in expressions like $\mathrm{poly}(\log q, d)$. What is meant is that there exist specific polynomials that can be used in the definition so that all subsequent statements in this paper hold.

The bound on ℓ is chosen to handle the length of the generator after reducing $\alpha\mathbb{O}$, which is $\mathrm{val}_\infty(\gamma/\alpha)$. The factor γ comes from ideal reduction, so γ has size polynomial in d, $\log q$, and $\mathrm{dg}\, N(\alpha)$.

Claim 4.5. *Given a BMR $(\beta_1, \dots, \beta_\ell)$ of a minimum β of $\mathbb{O}$, the ideal $(1/\beta)\mathbb{O}$ can be efficiently computed.*

Proof. At the first step, the ideal $(1/\beta_1)\mathbb{O}$, which is reduced by the definition of a BMR, can be efficiently computed. In general, let $\beta_i' = \prod_{j=1}^{i} \beta_j^{2^{i-j}}$. By the definition of a BMR, β_i' is a minimum of $\mathbb{O}$ for all i. Given the reduced ideal $(1/\beta_i')\mathbb{O}$, the reduced ideal $(1/\beta_{i+1}')\mathbb{O} = (1/(\beta_{i+1}\beta_i'^2))\mathbb{O}$ can be efficiently computed by squaring $(1/\beta_i')\mathbb{O}$ and multiplying by $1/\beta_{i+1}$. □

Our next algorithm produces a compact representation of a generator of an ideal. It calls Close (Algorithm 4.8), which calls Double (Algorithm 4.10); we will postpone the description of these algorithms, and the proofs of their correctness, until after the proof that Algorithm 4.6 is correct.

Algorithm 4.6 (Compact representation).

Input: A vector $v \in \mathbb{Z}^n$ and an ideal A such that $v = \mathrm{val}_\infty(\alpha)$ and $A = \alpha\mathbb{O}$ for some $\alpha \in K$.

Output: A compact representation of such an α.

1. Call Reduce$(A, 0)$ to get a reduced ideal $(1/\gamma)A$ and element $\gamma \in K$.
2. Let $(\beta_1, \dots, \beta_\ell) = \mathrm{Close}(\mathbb{O}, \mathrm{val}_\infty(\gamma/\alpha))$.
3. Output $(\gamma, (\beta_1, \dots, \beta_\ell))$.

Lemma 4.7. *Algorithm 4.6 returns a compact representation of* $\alpha \in K$ *in time polynomial in* $\log q$, d, $\mathrm{dg}\, N(\alpha)$, *and* $\log(\|\mathrm{val}_\infty(\alpha)\|_\infty)$.

Proof. Proposition 3.5 and Remark 3.8 show that the element γ in Step 1 can be computed with Algorithm 3.4 in time polynomial in d, $\log q$, and $\mathrm{dg}\, N(\alpha)$. Therefore the size of γ is bounded by the same amount. Also, γ is a minimum of $A = \alpha\mathbb{O}$, so $\beta := \gamma/\alpha$ is a minimum of $\mathbb{O}$. By Corollary 4.13 below, $\mathrm{Close}(\mathbb{O}, \mathrm{val}_\infty(\gamma/\alpha))$ returns the BMR $(\beta_1, \dots, \beta_\ell)$ of the minimum $\beta = \gamma/\alpha$ of $\mathbb{O}$ (and not just the BMR of a minimum close to γ/α). Hence the algorithm computes the compact representation $(\gamma, (\beta_1, \dots, \beta_\ell))$ of $\alpha = \gamma/\beta$.

We have already noted that Step 1 takes time polynomial in d, $\log q$, and $\mathrm{dg}\, N(\alpha)$. In Step 2, Algorithm Close is called, which executes $\ell = \log(\|\mathrm{val}_\infty(\gamma/\alpha)\|_\infty) + 1$ calls of Algorithm Double. Therefore it suffices to show that Double takes time polynomial in d, $\log q$, and $\mathrm{dg}\, N(\alpha)$.

Each call of Double calls Reduce once on input of the form $(B, \lfloor u \rceil)$; here B is the square of a reduced ideal, u is a vector in $\mathbb{Q}^k$ for some $k \le \ell$, with ℓ as above, and where $\lfloor u \rceil$ denotes the nearest integer vector to u. The ideal B is the square of a reduced ideal, and so is small. On the other hand, u is obtained from doubling a vector $t - \mathrm{val}_\infty(\mu)$ with $\|t - \mathrm{val}_\infty(\mu)\|_1 \le n + g$, so $\|u\|_1 \le 2n + 2g$. Rounding u to $\lfloor u \rceil$ adds at most $k/2$ to the 1-norm, and $k \le n$, so $\|\lfloor u \rceil\|_1 \le 5n/2 + 2g$. By Proposition 3.5, we find that each call of Reduce takes time polynomial in d, $\log q$, $\mathrm{dg}\, N(\alpha)$, as we were to show. □

Algorithm 4.8 (Close).

Input: A reduced ideal A and a vector $s \in \mathbb{Q}^n$.

Output: A BMR $(\beta_1, \dots, \beta_\ell)$ of a minimum $\beta \in A$ which is close to s, where $\ell = \log(\|s\|_\infty) + 1$.

```
1. Let $\beta_0 = 1$, $\ell = \log(\|s\|_\infty) + 1$ and $t = s/2^\ell$.
2. For $k$ from 1 to $\ell$
   (a) Let $(\beta_1, \dots, \beta_k) := \mathrm{Double}(A, t, (\beta_0, \beta_1, \dots, \beta_{k-1}))$.
   (b) Let $t := 2t$.
3. Return $(\beta_1, \dots, \beta_\ell)$.
```

Proposition 4.9. *Algorithm 4.8 is correct.*

Proof. This follows from Proposition 4.11, together with the fact that in Step 1 of the algorithm $\beta_0 = 1$ is a minimum of A which is close to $t = s/2^\ell$. □

Algorithm 4.10 (Double).

Input: A reduced ideal A, a vector $t \in \mathbb{Q}^n$, and a BMR $(\beta_1, \dots, \beta_{k-1})$ of a minimum β of A which is close to t.

Output: A BMR $(\beta_1, \dots, \beta_{k-1}, \beta_k)$ of a minimum of A which is close to $2t$, where β_k is a minimum of $(1/\beta^2)A$ that has size polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div}\, A)$.

```
1. Let $B := (1/\beta^2)A$ and $u := 2t - \mathrm{val}_\infty(\beta^2)$.
2. Reduce $(B, \lfloor u \rceil)$ to get a minimum $\beta_k$ of $B$ that is close to $u$. (Here $\lfloor u \rceil$ denotes
   the integer vector closest to $u$.)
3. Return $(\beta_1, \dots, \beta_{k-1}, \beta_k)$. (This is a BMR of $\beta^2 \cdot \beta_k$.)
```

Proposition 4.11. *Algorithm 4.10 is correct.*

Proof. First we show that there exists a minimum β_k of B such that

$$\|\mathrm{val}_\infty(\beta_k) - u\|_1 \le n/2 + g.$$

When we reduce the pair $(B, \lfloor u \rceil)$ we get a pair

$$((1/\beta_k)B, \lfloor u \rceil - \mathrm{val}_\infty(\beta_k)) \in \mathrm{Rep}^{f*}(B).$$

Let $t = (t_1, \dots, t_n) = \lfloor u \rceil - \mathrm{val}_\infty(\beta_k)$. By Proposition 3.3, we have

$$\sum_{i=1}^{n} t_i \deg \mathfrak{p}_i \le g,$$

where g is the genus of the function field K. The difference between the ℓ_1-norms of u and $\lfloor u \rceil$ is at most $n/2$, so $\mathrm{val}_\infty(\beta_k) - u$ has ℓ_1-norm bounded by $n/2 + g$. Thus there exists a minimum β_k of B that is close to $u = 2t - \mathrm{val}_\infty(\beta^2)$, and this minimum is computed in Step 3 of Double. Moreover, by Proposition 3.5, the size of the minimum β_k is polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div}(B))$, and $\|u\|_\infty$. Then, since β_k is close to $2t - \mathrm{val}_\infty(\beta^2)$, we have that $\beta^2\beta_k$ is close to $2t$, because

$$\|2t - \mathrm{val}_\infty(\beta^2\beta_k)\|_1 = \|(2t - (\mathrm{val}_\infty(\beta^2)) - \mathrm{val}_\infty(\beta_k)\|_1. \qquad \square$$

In the next proposition we show that if there is a minimum of A whose valuation vector equals $2t$, then Double returns a BMR of this minimum.

Proposition 4.12. *Let A be a reduced ideal. Suppose there is a minimum μ of A such that* $\mathrm{val}_\infty(\mu) = 2t$. *Then* Double$(A, t, \beta = (\beta, \dots, \beta_{k-1}))$ *returns the BMR $(\beta_1, \dots, \beta_k)$ of this minimum*; *that is, $\mu = \beta^2\beta_k$.*

Proof. In Step 3 of Double the algorithm reduces the pair $((1/\beta^2)A, 2t - \mathrm{val}_\infty(\beta^2))$, where β is the given minimum of A which is close to t. Since $2t = \mathrm{val}_\infty(\mu)$, we see that $2t$ has integer coordinates, so it is not necessary to round $u = 2t - \mathrm{val}_\infty(\beta^2)$.

After reducing $((1/\beta^2)A, 2t - v(\beta^2))$ we obtain an element

$$((1/(\beta_k\beta^2))A, 2t - \mathrm{val}_\infty(\beta^2) - \mathrm{val}_\infty(\beta_k))$$

of $\mathrm{Rep}^{f*}(\mathbb{O})$, where β_k is a minimum of $(1/\beta^2)A$. Since reduction produces a unique element in $\mathrm{Rep}^{f*}(\mathbb{O})$ and elements of $\mathrm{Rep}^{f*}(\mathbb{O})$ have unique representatives, this implies that β_k is uniquely determined (up to multiplication by an element of $\mathbb{F}_q^*$). Since μ is a minimum of A, we have $((1/\mu)A, 0) \in \mathrm{Rep}^{f*}(\mathbb{O})$. We also have that $\nu := \mu/(\beta^2)$ is a minimum of $(1/\beta^2)A$. Then

$$\big((1/\nu)(1/\beta^2)A, 2t - \mathrm{val}_\infty(\beta^2) - \mathrm{val}_\infty(\nu)\big) = ((1/\mu)A, 0) \in \mathrm{Rep}^{f*}(\mathbb{O}).$$

Hence we must have $\beta_k = \mu/\beta^2$; that is, Double returns a BMR of $\mu = \beta_k\beta^2$. $\square$

Corollary 4.13. *If the input in* Close (*Algorithm 4.8*) *is a reduced ideal A and a vector $s \in \mathbb{Q}^n$ such that $s =$* $\mathrm{val}_\infty(\mu)$ *for a minimum μ of A, then* Close *outputs a BMR of μ.*

Proof. At the last step of the for loop in Step 2 of Close, we have a BMR of a minimum β of A that is close to $s/2$, and the last call of Double produces a BMR of a minimum β' of A that is close to s. By Proposition 4.12, Double outputs a BMR of μ, so Algorithm Close returns a BMR of μ with $s = \mathrm{val}_\infty(\mu)$. $\square$

Corollary 4.14. *The principal ideal problem is in NP.*

Proof. Given a function field and an ideal I of $\mathbb{O}$ represented in Hermite normal form (HNF), if the ideal is principal, then the proof is a compact representation

$B = (\gamma, (\beta_1, \dots, \beta_\ell))$ for α, where $I = \alpha\mathbb{O}$. By Definition 4.3, the compact representation B has size bounded by $\log(\|\mathrm{val}_\infty(\alpha)\|_\infty + g)$ and $\mathrm{poly}(\log q, d, \deg N(\alpha))$. The field parameters are $\log q$, d, and g. By Remark 3.8, being polynomial in $\deg(N(\alpha))$ is the same as being polynomial in $\mathrm{ht}(\mathrm{div}(A))$, which is the size of the ideal $A = \alpha\mathbb{O}$. Propositions 3.3 and 3.5 tell us that $\|\log \mathrm{val}_\infty(\alpha)\|_\infty$ is bounded.

The verifier must efficiently test whether $A = (\gamma/\beta)\mathbb{O}$, where $\beta = \prod \beta_i^{2^{n-i}}$. The verifier can efficiently compute the ideal as follows. By Claim 4.5, $(1/\beta)\mathbb{O}$ can be efficiently computed. Multiplication by γ is efficient. Finally, comparing the HNF of A and $(\gamma/\beta)\mathbb{O}$ is efficient since the representation of an ideal is unique. □

5. Quantum algorithms for the unit group, the principal ideal problem, and the class group

In this section we give efficient quantum algorithms for computing the unit group, solving the principal ideal problem and computing the class group. Recall from Section 3 that for the unit group and the principal ideal problem this means the objects are computed in the val_∞ embedding, and that compact representations can then be computed.

The basic approach is to show that each of these problems reduces to an instance of the abelian hidden subgroup problem (HSP), which is known to have an efficient quantum algorithm [10]. The class group case is slightly more general since the HSP instances will take values that are quantum states.

Theorem 5.1. *There is a polynomial-time quantum algorithm for computing the unit group of a function field.*

Proof. A hidden subgroup problem for the unit group can be defined by the function $g \colon \mathbb{Z}^n \to \mathrm{Rep}^{f*}(\mathbb{O})$ defined as $g(e) = \mathrm{Reduce}(\mathbb{O}, e)$. Here $\mathrm{Reduce}(\mathbb{O}, e)$ is the element of $\mathrm{Rep}^{f*}(\mathbb{O})$ that is computed by Algorithm 3.4; it is polynomial-time computable by Proposition 3.5. By Remark 3.6,

$$\mathrm{Reduce}(\mathbb{O}, e) = \mathrm{Reduce}(\mathbb{O}, e + v)$$

for every $v \in \Lambda$, so the function g is constant on cosets. Therefore g is also defined on $\mathbb{Z}^n/\Lambda$, and it gives a bijection between $\mathbb{Z}^n/\Lambda$ and $\mathrm{Rep}^{f*}(\mathbb{O})$, so it is also distinct on different cosets. Using the HSP instance g, a quantum algorithm can compute a basis for Λ efficiently. Compact representations can then be computed if desired. □

In the *decision* version of the principal ideal problem, an ideal I of $\mathbb{O}$ is given in HNF and the problem is to decide if it is principal. If it is principal, then the *search* version of the problem is to compute a generator; that is, to compute an α such that $I = \alpha\mathbb{O}$. Since generators may take an exponential number of bits to represent in general, we only require computing $\mathrm{val}_\infty(\alpha)$. Knowing $\mathrm{val}_\infty(\alpha)$ and $\alpha\mathbb{O}$ determines α up to multiplication by an element of k^*. So given an arbitrary ideal I that is given to us in HNF, the strategy is to solve the search problem and compute a candidate value for $\mathrm{val}_\infty(\alpha)$, and then to test whether $I = \alpha\mathbb{O}$ to see if the ideal is principal or not. A compact representation of α can then be computed from $\mathrm{val}_\infty(\alpha)$ and I using Algorithm 4.6.

Theorem 5.2. *There is a polynomial-time quantum algorithm for the principal ideal problem in a function field.*

Proof. Recall that for a vector $v \in \mathbb{Z}^n$, calling Algorithm 3.4 on $(\mathbb{O}, v)$ results in a pair $(I_v, f_v) \in \mathrm{Rep}^{f*}(\mathbb{O})$. Here I_v is a reduced ideal and f_v is a vector such that f_v has positive coordinates. If $(1/\mu)\mathbb{O} = I_v$ then $\mathrm{val}_\infty(1/\mu) + f_v = v$ by Remark 3.6.

To solve the principal ideal problem we do the following: Given any ideal I we first call Algorithm 3.4 on $(I, 0)$ to get a reduced ideal I_v. The reduction algorithm also computes γ such that $(1/\gamma)I = I_v$. Hence it suffices to solve the principal ideal problem for reduced ideals. If $I_v = (1/\mu)\mathbb{O}$ is reduced, then I_v represents the point $v + \Lambda \in \mathbb{Z}^n/\Lambda$ with $v = \mathrm{val}_\infty(\mu)$. By the above discussion, solving the principal ideal problem means computing v. First, by Theorem 5.1, a basis B of the unit group (under the embedding val_∞) can be computed efficiently with a quantum algorithm. A hidden subgroup problem can be set up as follows. By abuse of notation we denote by $\mathbb{Z}^n/B$ the quotient of $\mathbb{Z}^n$ by the lattice generated by the elements in B. Let $G = \mathbb{Z}_M \times \mathbb{Z}^n/B$, where $M = |\mathbb{Z}^n/B|$. Define $h\colon G \to \mathrm{Rep}^{f*}(K) = \bigcup_A \mathrm{Rep}^{f*}(A)$ by the following algorithm: On input (a, b), use the composition operation in Section 3C and repeated doubling to compute a times the group element (this does reductions along the way, giving an element in $\mathrm{Rep}^{f*}(K)$); then compose the result with $(\mathbb{O}, -b)$ and reduce. When the ideal I is principal, we have $h(a, b) = (I_{av-b}, f_{av-b})$. The hidden subgroup in this case is $H = \langle(1, v)\rangle$, and $h(H) = (\mathbb{O}, 0)$. A set of coset representatives for H is $\{(0, w) : w \in \mathbb{Z}^n/B\}$. Then $h((0, w) + n(1, v)) = h(n, w + nv) = (I_{-w}, f_{-w})$, and so the different values of w correspond to the set of elements in $\mathrm{Rep}^{f*}(\mathbb{O})$. So h is constant on cosets and distinct on different cosets. The function h can be computed efficiently using a small modification of Close (Algorithm 4.8). Therefore there is an efficient quantum algorithm for finding generators for H. Given an element $(n, nv) \in \mathbb{Z}_M \times \mathbb{Z}^n/B$ of H, it is easy to compute v. □

Theorem 5.3. *There is a polynomial-time quantum algorithm for computing the ideal class group of a function field.*

Proof. To compute the class group we also reduce to an abelian hidden subgroup problem where the function takes quantum states as values. Since it is not known how to compute unique representatives in the class group we instead create quantum states to represent each element, as a superposition over all elements of

$\mathrm{Rep}^{f*}(J)$ for the ideal class of J. Let $g_1, \dots, g_m$ be a set of generators for $\mathrm{Cl}(\mathbb{O})$; Appendix A shows how to compute such a set. For an ideal J, let

$$|\phi_J\rangle = \sum_{(I,v)\in \mathrm{Rep}^{f*}(J)} |I, v\rangle.$$

Define

$$f \colon \mathbb{Z}^m \to \mathbb{C}^{|\operatorname{Pic}^0(K)|}$$

by $f(e_1, \dots, e_m) = |\phi_J\rangle$, where J is the ideal resulting from $\mathrm{Reduce}(g_1^{e_1} \cdots g_m^{e_m}, 0)$. The function f can be efficiently evaluated using the algorithm for the principal ideal problem as follows. Given $|e_1, \dots, e_m\rangle$, compute $|e_1, \dots, e_m, J\rangle$, where J is the ideal resulting from $\mathrm{Reduce}(g_1^{e_1} \cdots g_m^{e_m}, 0)$. The ideal in the last register, call it J, is now used to create the superposition over reduced ideals. Create $\sum_{v\in\mathbb{Z}^n/B} |J, v\rangle$, then $\sum_{v\in\mathbb{Z}^n/B} |J, v, (J_v, f_v)\rangle$ where (J_v, f_v) is the result of calling $\mathrm{Reduce}(J, v)$. Next use the principal ideal algorithm on $J \cdot J_v^{-1}$, which outputs v, to create $\sum_{v\in\mathbb{Z}^n/B} |J, v, (J_v, f_v), v\rangle$. Next uncompute v in the second register using the fourth, then uncompute the fourth register by running the principal ideal algorithm backwards. Finally, uncompute J using $e_1, \dots, e_m$. □

Acknowledgments

This work was supported in part by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-08-1-0298. The first author was partially supported by National Science Foundation grant DMS-1056703 and a Sloan Research Fellowship. The second author was partially supported by National Science Foundation grant CCF-0747274.

Appendix A. Computing generators for Cl($\mathbb{O}$)

As usual, let K be an algebraic function field over a finite field of constants $k = \mathbb{F}_q$. As discussed in Section 2, when $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_{n+1}\}$ is the set of places at infinity and $\deg \mathfrak{p}_{n+1} = 1$, we have a short exact sequence

$$0 \longrightarrow \mathrm{Ker} \longrightarrow \operatorname{Pic}^0(K) \longrightarrow \mathrm{Cl}(\mathbb{O}) \to 1$$

where the map from $\operatorname{Pic}^0(K) \to \mathrm{Cl}(\mathbb{O})$ is given as

$$\sum_{\mathfrak{p}\in\mathcal{P}_K} n_{\mathfrak{p}}\mathfrak{p} \longmapsto \prod_{\mathfrak{p}\in\mathcal{P}_K - S} (\mathfrak{p} \cap \mathbb{O})^{-n_{\mathfrak{p}}}.$$

Given a function field K as above, there is a smooth projective geometrically irreducible curve C whose function field is K. Let g denote the genus of this curve.

In [28] Kedlaya proved that for q with $q^{1/2} > 16g$ there exists a randomized algorithm that produces a generating set for $\operatorname{Pic}^0(K)$ in time polynomial in g and

$\log q$. The genus of the curve C does not change if we increase the size of the base field k. Hence by enlarging the constant field, if necessary, we may assume that $q^{1/2} > 16g$. From the exact sequence above it follows that the image of the generating set for $\mathrm{Pic}^0(K)$ under the map described above gives a generating set of $\mathrm{Cl}(\mathcal{O})$.

Appendix B. Computing Riemann-Roch spaces

In this section we analyze the complexity of computing the Riemann-Roch space $L(D) := \{f \in K : \mathrm{div}(f) + D \geq 0\} \cup \{0\}$. The input to the problem is a function field K and a divisor

$$D = \left(A, \sum_{i=1}^{n+1} t_i \mathfrak{p}_i\right)$$

of K. The fractional ideal A of $\mathcal{O}$ is given in HNF relative to an $\mathcal{O}$-basis. The second part of D is given by a set of integers $\{t_i : 1 \leq i \leq n+1\}$ that determine the multiplicity of the infinite places, that is, the places of $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$, in D.

We follow Hess's [26] algorithm to compute the Riemann-Roch space. In [26] Hess does not include any proofs for the complexity of his algorithm, so in this section we show that the Riemann-Roch space $L(D)$ can be computed in time polynomial in d, $\log q$ and $\mathrm{ht}(D)$. (For the definition of $\mathrm{ht}(D)$ see Section 2.) Hess's algorithm is a relatively simple, self-contained algorithm. We also investigate more closely the complexity of computing $\mathfrak{o}_\infty$-bases of the ideals we are working with.

The main idea in [26] is that the Riemann-Roch space can be computed as the intersection of two ideals that come from the divisor D, where the two ideals are in the rings $\mathcal{O}$ and $\mathcal{O}_\infty$.

First we show that we can compute an $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$ in polynomial time.

Proposition B.1 ([15], Proposition 4.13). *Let $R \subset S$ be commutative rings with identity and let U be a multiplicatively closed subset of R. If S' is the integral closure of R in S, then $S'[U^{-1}]$ is the integral closure of $R[U^{-1}]$ in $S[U^{-1}]$.*

Lemma B.2. *An $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$ is computable in time polynomial in d and* $\log q$.

Proof. By [11, Theorem 1] applied to $k[1/x]$, we can compute a basis $\beta_1, \ldots, \beta_d$ of the integral closure of $k[1/x]$ in K. By Proposition B.1, taking integral closures commutes with localization, so when we apply the proposition to the rings $R = k[1/x]$ and $S = K$, with U being the complement of the prime ideal $(1/x)$ of R, we find that $\mathfrak{o}_\infty = k[1/x][U^{-1}]$. Let S' be the integral closure of $k[1/x]$ in K. Then $\mathcal{O}_\infty = S'[U^{-1}]$, which implies that $\beta_1, \ldots, \beta_d$ is an $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$. □

Lemma B.3. *Let A be a fractional ideal of $\mathcal{O}$ given by a $k[x]$-basis, and let B be a fractional ideal of $\mathcal{O}_\infty$ given by an $\mathfrak{o}_\infty$-basis. There exist bases $a_1, \ldots, a_d$ of A and $b_1, \ldots, b_d$ of B and uniquely determined integers λ_i such that $x^{-\lambda_i} b_i = a_i$.*

Proof. Let $a'_1, \dots, a'_d \in K$ be a $k[x]$-basis of A and $b'_1, \dots, b'_d \in K$ a $\mathfrak{o}_\infty$-basis of B. Both of these are bases for K as a $k(x)$-vector space. Let $M \in k(x)^{d\times d}$ be such that

$$(a'_1, \dots, a'_d) = (b'_1, \dots, b'_d)M.$$

By [26, Corollary 4.3] there exists a unimodular $T_1 \in \mathfrak{o}_\infty^{d\times d} \subset k[[x^{-1}]]^{d\times d}$ and a unimodular $T_2 \in k[x]^{d\times d}$ such that $T_1 M T_2 = (x^{-\lambda_j}\delta_{ij})_{ij}$.

Let $(a_1, \dots, a_d) = (a'_1, \dots, a'_d)T_2$ and $(b_1, \dots, b_d) = (b'_1, \dots, b'_d)T_1^{-1}$. Then

$$(b_1, \dots, b_n)T_1 M T_2 = (b'_1, \dots, b'_d)MT_2 = (a'_1, \dots, a'_d)T_2 = (a_1, \dots, a_d). \quad \square$$

Lemma B.4. *If $a_1, \dots, a_d$ and $b_1, \dots, b_d$ are bases as in Lemma B.3, then $A \cap B$ has k-basis $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_j\}$.*

Proof. Assume $\lambda \ge 0$. Because $x \in \mathcal{O}$, the elements $x^j a_i$ lie in A for $j \ge 0$, so all we have to show is that $x^j a_i \in B$ if and only if $0 \le j \le \lambda$. We have $a_i = x^{-\lambda_i} b_i \in B$ since $1/x \in \mathfrak{o}_\infty$, B is an $\mathfrak{o}_\infty$-module and $\lambda_i \ge 0$. Similarly, $x^j a_i = x^{j-\lambda_i} b_i \in B$ if and only if $j - \lambda_i \le 0$, that is, if and only if $j \le \lambda_i$. But $x^j a_i \in A$ if and only if $j \ge 0$, so $x^j a_i \in A \cap B$ for $0 \le j \le \lambda_i$.

To see that this set forms a k-basis note that $A \cap B = \bigcup_{i=1}^d (A \cap B \cap k(x)a_i)$, and a k-basis for $A \cap B$ is the union of the k-bases for $A \cap B \cap k(x)a_i$.

But for i with $\lambda_i \ge 0$ we have $A \cap B \cap k(x)a_i = A \cap B \cap a_i k[x]$, so it suffices to determine which monomials $(x^j)a_i$ are in this intersection. By the above argument the only monomials in this intersection are $a_i, xa_i, \dots, x^{\lambda_i}a_i$, and these elements are clearly linearly independent over k, so they form a k-basis for $A \cap B \cap k(x)a_i$ (for i with $\lambda_i \ge 0$). □

Lemma B.5. *The elements $a_1, \dots, a_d$ and the integers $\lambda_1, \dots, \lambda_d$ above can be computed in polynomial time.*

Proof. We will first show that the matrices M and T_2 from the proof of Lemma B.3 can be computed in polynomial time. The lemma then follows from the fact that $(a_1, \dots, a_d) = (a'_1, \dots, a'_d)T_2$, and that the maximum degree of elements of the j-th column of MT_2 is equal to $-\lambda_j$([19, p. 15], [26, Corollary 4.3]). When elements in K are specified as polynomials in y, that is, as $\sum_{i=0}^n a_i y^i$ for coefficients $a_i \in k(x)$, then writing a element $\alpha \in K$ in terms of a basis $\omega_1, \dots, \omega_n$ is a vector space transformation, with vector space generators $1, y, y^2, \dots, y^{n-1}$. The columns of the matrix $A \in k(x)^{n\times n}$ contain the coefficients of the polynomials $\omega_1, \dots, \omega_n$. Then solving the equation $Az = b$ over $k(x)$ for z gives the coefficients of b in terms of the basis. For M, this can be done for each column.

The matrix T_2 is computed using Paulus's polynomial-time algorithm [32] by keeping track of the operations during the basis reduction. □

Algorithm B.6 (Ideal intersection for ideals in two different rings).

Input: A function field K; an element $x \in K$; a $k[x]$-basis $\omega_1, \dots, \omega_d$ of $\mathcal{O}$; a $k[x]$-basis $a'_1, \dots, a'_d$ of the fractional ideal A of $\mathcal{O}$; an $\mathfrak{o}_\infty$-basis $v_1, \dots, v_d$ of $\mathcal{O}_\infty$; and an $\mathfrak{o}_\infty$-basis $b'_1, \dots, b'_d$ of the fractional ideal B of $\mathcal{O}_\infty$.

Output: Elements $a_1, \dots, a_d$ of K and integers $\lambda_1, \dots, \lambda_d$ such that $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ is k-basis of the k-vector space $A \cap B$.

1. Compute a matrix M such that $(b'_1, \dots, b'_d)M = (a'_1, \dots, a'_d)$.
2. Do a basis reduction on M. Keep track of the operations and let $T_2 \in Gl_d(k[x])$ be the transformation. Let $-\lambda_i$ be the maximum degree in the i-th column of the reduced matrix MT_2.
3. Let $(a_1, \dots, a_d) = (a'_1, \dots, a'_d)T_2$.
4. Return $(a_1, \dots, a_d; \lambda_1, \dots, \lambda_d)$.

Proposition B.7. *Algorithm B.6 is correct, and runs in polynomial time.*

Proof. The matrix M computed in Step 1 of the algorithm is exactly the matrix from Lemma B.3 that leads to the special basis for A; that is, $(a_1, \dots, a_d) = (a'_1, \dots, a'_d)T_2$. By Lemma B.4 and its proof, if $-\lambda_j$ is the maximum column degree in the j-th column of MT_2, then $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ is a k-basis for the intersection $A \cap B$. By Lemma B.5, the a_i and the λ_i can be computed in polynomial time. □

Algorithm B.8 (Riemann-Roch space).

Input: A function field K; a $k[x]$-basis $\omega_1, \dots, \omega_d$ of $\mathcal{O}$; and a divisor $D = (A, \sum_{i=1}^{n+1} t_i \mathfrak{p}_i)$, where A is a fractional ideal of $\mathcal{O}$ given in a $k[x]$-basis.

Output: Elements $a_1, \dots, a_d$ of K and integers $\lambda_1, \dots, \lambda_d$ such that $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ is a basis of the Riemann-Roch space $L(D)$.

1. Compute a $k[x]$-basis of A^{-1}.
2. Compute an $\mathfrak{o}_\infty$-basis of $B := \Pi_{i=1}^{n+1} (\mathfrak{p}_i \cap \mathcal{O}_\infty)^{t_i} \subseteq \mathcal{O}_\infty$.
3. Compute an $\mathfrak{o}_\infty$-basis of B^{-1}.
4. Use Algorithm B.6 to compute $A^{-1} \cap B^{-1}$.
5. Return the $(a_1, \dots, a_d; \lambda_1, \dots, \lambda_d)$ computed by Algorithm B.6.

Theorem B.9. *Algorithm B.8 computes the Riemann-Roch space $L(D)$ in time polynomial in d, $\log q$, and $\mathrm{ht}(D)$.*

Proof. Computing the inverse of a fractional ideal A of $\mathbb{O}$ can be done in time polynomial in $\log q$, d, and $\mathrm{ht}(\mathrm{div}(A))$ [14, Proposition 2.69(b)]. The ideals $\mathfrak{p}_i \cap \mathbb{O}_\infty$ in Step 2 are the prime ideals of $\mathbb{O}_\infty$ corresponding to the places in S. These can be computed in polynomial time with an algorithm similar to the one given for number fields in [16]. Hence we can compute an $\mathfrak{o}_\infty$-basis for the ideal B in Step 2 in polynomial time. The inverse of an ideal B in this ring can be computed efficiently as well. Finally, by Proposition B.7 above, a basis for the k-vector space $A^{-1} \cap B^{-1}$ can be computed in polynomial time. □

References

[1] ACM (ed.), *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, held in Baltimore, MD, May 22–24, 2005*, New York, Association for Computing Machinery, 2005. MR 2006f:68006

[2] ACM (ed.), *STOC'07—Proceedings of the 39th Annual ACM Symposium on Theory of Computing, held in San Diego, CA, June 11–13, 2007*, New York, Association for Computing Machinery, 2007. MR 2009a:68002

[3] ACM (ed.), *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing, held in Bethesda, MD, May 31–June 2, 2009*, New York, Association for Computing Machinery, 2009. MR 2012d:68003

[4] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh Huang, *A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields*, in Adleman and Huang [5], 1994, pp. 28–40. MR 96b:11078

[5] Leonard M. Adleman and Ming-Deh Huang (eds.), *Algorithmic number theory: Proceedings of the First International Symposium (ANTS-I) held at Cornell University, Ithaca, New York, May 6–9, 1994*, Lecture Notes in Computer Science, no. 877, Berlin, Springer, 1994. MR 95j:11119

[6] G. Brassard (ed.), *Advances in cryptology—CRYPTO '89: Proceedings of the Conference on the Theory and Applications of Cryptology held at the University of California, Santa Barbara, California, August 20–24, 1989*, Lecture Notes in Computer Science, no. 435, New York, Springer, 1990. MR 91b:94002

[7] Johannes A. Buchmann and Hugh C. Williams, *A key exchange system based on real quadratic fields (extended abstract)*, in Brassard [6], 1990, pp. 335–343. MR 91f:94014

[8] J. P. Buhler (ed.), *Algorithmic number theory: Proceedings of the 3rd International Symposium (ANTS-III) held at Reed College, Portland, OR, June 21–25, 1998*, Lecture Notes in Computer Science, no. 1423, Berlin, Springer, 1998. MR 2000g:11002

[9] Moses Charikar (ed.), *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms held in Austin, TX, January 17–19, 2010*, Philadelphia, PA, Society for Industrial and Applied Mathematics, 2010. MR 2012f:68008

[10] Kevin K. H. Cheung and Michele Mosca, *Decomposing finite abelian groups*, Quantum Inf. Comput. **1** (2001), no. 3, 26–32. MR 2003e:81030

[11] A. L. Chistov, *The complexity of the construction of the ring of integers of a global field*, Dokl. Akad. Nauk SSSR **306** (1989), no. 5, 1063–1067. MR 90g:11170

[12] Henri Cohen (ed.), *Algorithmic number theory: Proceedings of the 2nd International Symposium (ANTS-II) held at the Université Bordeaux I, Talence, May 18–23, 1996*, Lecture Notes in Computer Science, no. 1122, Berlin, Springer, 1996. MR 97k:11001

[13] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren (eds.), *Handbook of elliptic and hyperelliptic curve cryptography*, Chapman & Hall/CRC, Boca Raton, FL, 2006. MR 2007f:14020

[14] Claus Diem, *On arithmetic and the discrete logarithm problem in class groups of curves*, Habilitationsschrift, Universität Leipzig, 2008. http://www.math.uni-leipzig.de/∼diem/preprints/habil.pdf

[15] David Eisenbud, *Commutative algebra, with a view toward algebraic geometry*, Graduate Texts in Mathematics, no. 150, Springer, New York, 1995. MR 97a:13001

[16] Kirsten Eisenträger and Sean Hallgren, *Algorithms for ray class groups and Hilbert class fields*, in Charikar [9], 2010, pp. 471–483. MR 2012i:11103

[17] Felix Fontein, *The infrastructure of a global field of arbitrary unit rank*, Math. Comp. **80** (2011), no. 276, 2325–2357. MR 2012f:11243

[18] Felix Fontein and Pawel Wocjan, *Quantum algorithm for computing the period lattice of an infrastructure*, 2011. arXiv 1111.1348 [quant-ph]

[19] Felix Wolfgang Fontein, *The infrastructure of a global field and baby step-giant step algorithms*, Ph.D. thesis, University of Zurich, 2009. http://dx.doi.org/10.5167/uzh-24993

[20] Arnaldo García and Henning Stichtenoth, *A tower of Artin-Schreier extensions of function fields attaining the Drinfel′d-Vlăduţ bound*, Invent. Math. **121** (1995), no. 1, 211–222. MR 96d:11074

[21] Henri Gilbert (ed.), *Advances in cryptology—EUROCRYPT* 2010*: Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques held on the French Riviera, May* 30*–June* 3*,* 2010, Lecture Notes in Computer Science, no. 6110, Berlin, Springer, 2010. MR 2011g:94001

[22] V. D. Goppa, *Geometry and codes*, Mathematics and its Applications (Soviet Series), no. 24, Kluwer Academic Publishers Group, Dordrecht, 1988. MR 91a:14013

[23] Venkatesan Guruswami, *Constructions of codes from number fields*, IEEE Trans. Inform. Theory **49** (2003), no. 3, 594–603. MR 2004g:94093

[24] ———, *Artin automorphisms, cyclotomic function fields, and folded list-decodable codes* (*extended abstract*), in ACM [3], 2009, pp. 23–32. MR 2780046

[25] Sean Hallgren, *Fast quantum algorithms for computing the unit group and class group of a number field*, in ACM [1], 2005, pp. 468–474. MR 2006g:81032

[26] F. Hess, *Computing Riemann-Roch spaces in algebraic function fields and related topics*, J. Symbolic Comput. **33** (2002), no. 4, 425–445. MR 2003j:14032

[27] Ming-Deh Huang and Doug Ierardi, *Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve*, J. Symbolic Comput. **18** (1994), no. 6, 519–539. MR 96h:14077

[28] Kiran S. Kedlaya, *Quantum computation of zeta functions of curves*, Comput. Complexity **15** (2006), no. 1, 1–19. MR 2007b:14042

[29] A. K. Lenstra and H. W. Lenstra, Jr. (eds.), *The development of the number field sieve*, Lecture Notes in Mathematics, no. 1554, Springer, Berlin, 1993. MR 96m:11116

[30] Vadim Lyubashevsky, Chris Peikert, and Oded Regev, *On ideal lattices and learning with errors over rings*, in Gilbert [21], 2010, pp. 1–23. MR 2660480

[31] Volker Müller, Andreas Stein, and Christoph Thiel, *Computing discrete logarithms in real quadratic congruence function fields of large genus*, Math. Comp. **68** (1999), no. 226, 807–822. MR 99i:11119

[32] Sachar Paulus, *Lattice basis reduction in function fields*, in Buhler [8], 1998, pp. 567–575. MR 2000i:11193

[33] Chris Peikert and Alon Rosen, *Lattices that admit logarithmic worst-case to average-case connection factors*, in ACM [2], 2007, pp. 478–487. MR 2010e:68099

[34] Michael Rosen, *Number theory in function fields*, Graduate Texts in Mathematics, no. 210, Springer, New York, 2002. MR 2003d:11171

[35] Pradeep Sarvepalli and Pawel Wocjan, *Quantum algorithms for one-dimensional infrastructures*, 2011. arXiv 1106.6347 [quant-ph]

[36] R. Scheidler, *Compact representation in real quadratic congruence function fields*, in Cohen [12], 1996, pp. 323–336. MR 98c:11126

[37] ———, *Decision problems in quadratic function fields of high genus*, J. Complexity **16** (2000), no. 2, 411–423. MR 2001e:11112

[38] Arthur Schmidt and Ulrich Vollmer, *Polynomial time quantum algorithm for the computation of the unit group of a number field* (*extended abstract*), in ACM [1], 2005, pp. 475–480. MR 2006g:81038

[39] René Schoof, *Algebraic curves over* $\mathbf{F}_2$ *with many rational points*, J. Number Theory **41** (1992), no. 1, 6–14. MR 93h:11062

[40] Nigel P. Smart, *Reduced ideals in function fields*, Tech. Report, HP Laboratories Bristol, 1998. http://www.hpl.hp.com/techreports/98/HPL-98-201.html

[41] Henning Stichtenoth, *Algebraic function fields and codes*, 2nd ed., Graduate Texts in Mathematics, no. 254, Springer, Berlin, 2009. MR 2010d:14034

[42] Christoph Thiel, *On the complexity of some problems in algorithmic algebraic number theory*, Ph.D. thesis, Universität des Saarlandes, 1995. http://tinyurl.com/thiel-phd

[43] Emil J. Volcheck, *Computing in the Jacobian of a plane algebraic curve*, in Adleman and Huang [5], 1994, pp. 221–233. MR 96a:14033

KIRSTEN EISENTRÄGER: eisentra@math.psu.edu
Department of Mathematics, Penn State University, University Park, PA 16802, United States

SEAN HALLGREN: hallgren@cse.psu.edu
Department of Computer Science and Engineering, Penn State University,
University Park, PA 16802, United States

VOLUME EDITORS

Everett W. Howe
Center for Communications Research
4320 Westerra Court
San Diego, CA 92121-1969
United States

Kiran S. Kedlaya
Department of Mathematics
University of California, San Diego
9500 Gilman Drive #0112
La Jolla, CA 92093-0112

Front cover artwork based on a detail of
Chicano Legacy 40 Años ©2010 Mario Torero.

Electronic copies can be obtained free of charge from http://msp.org/obs/1 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-00-2 (print), 978-1-935107-01-9 (electronic)

First published 2013.

msp

MATHEMATICAL SCIENCES PUBLISHERS
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org http://msp.org

THE OPEN BOOK SERIES 1

Tenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the tenth ANTS meeting, held July 9–13, 2012, at the University of California, San Diego. It includes revised and edited versions of the 25 refereed papers presented at the conference, together with extended abstracts of two of the five invited talks.

TABLE OF CONTENTS