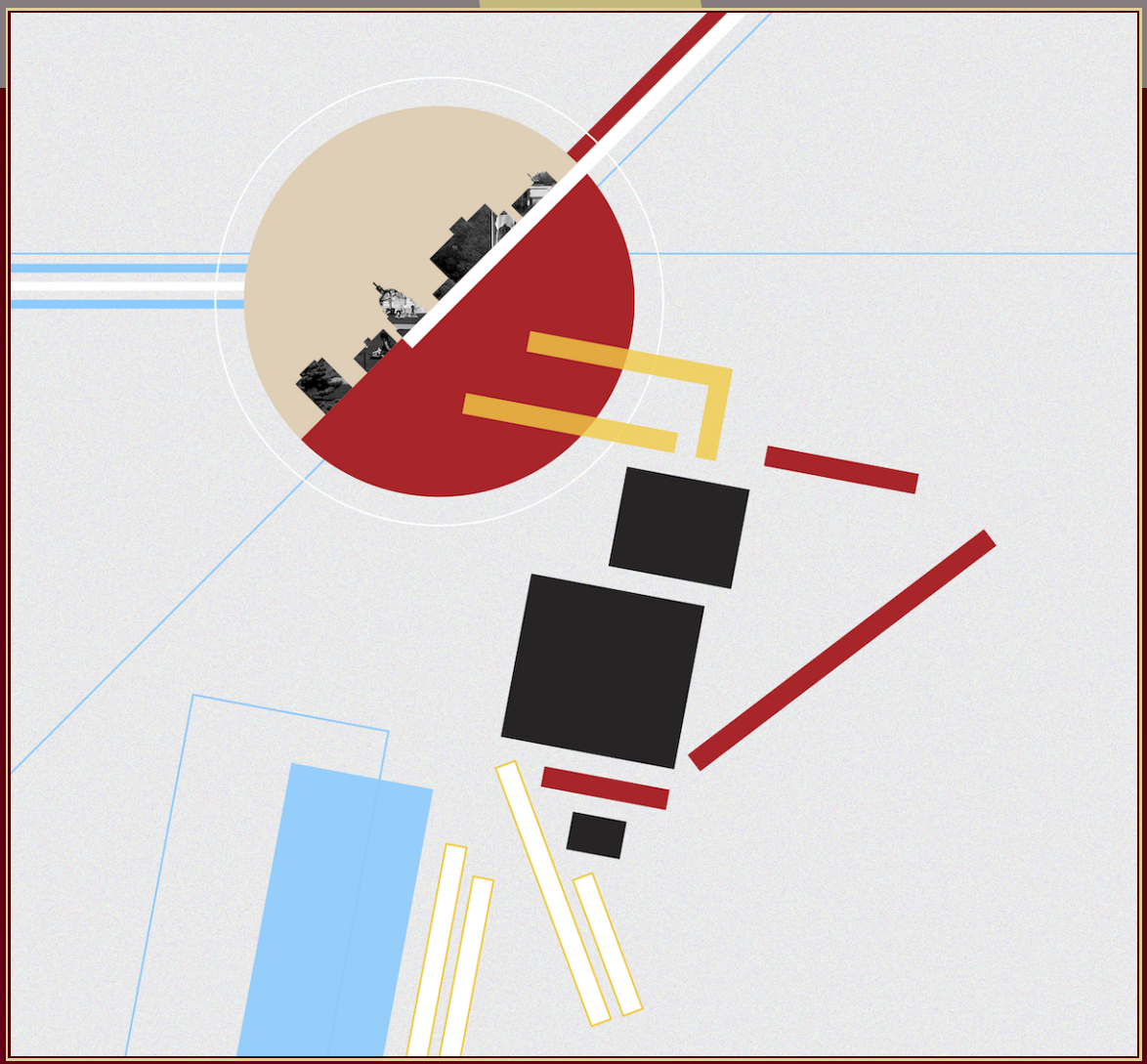
THE OPEN BOOK SERIES 2

ANTS XIII
Proceedings of the Thirteenth Algorithmic Number Theory Symposium

Fast multiquadratic S-unit computation and application to the calculation of class groups

Jean-François Biasse and Christine Van Vredendaal



msp

THE OPEN BOOK SERIES **2** (2019)
Thirteenth Algorithmic Number Theory Symposium
dx.doi.org/10.2140/obs.2019.2.103

msp

Fast multiquadratic S-unit computation and application to the calculation of class groups

Jean-François Biasse and Christine Van Vredendaal

Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a real multiquadratic field and S be a set of prime ideals of L. In this paper, we present a heuristic algorithm for the computation of the S-class group and the S-unit group that runs in time $\text{Poly}(\log(\Delta), \text{Size}(S))e^{\widetilde{O}(\sqrt{\ln d})}$ where $d = \prod_{i \leq n} d_i$ and Δ is the discriminant of L. We use this method to compute the ideal class group of the maximal order $\mathcal{O}_L$ of L in time $\text{Poly}(\log(\Delta))e^{\widetilde{O}(\sqrt{\log d})}$. When $\log(d) \leq \log(\log(\Delta))^c$ for some constant $c < 2$, these methods run in polynomial time. We implemented our algorithm using Sage 7.5.1.

1. Introduction

Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a real multiquadratic number field, and S be a set of prime ideals of L. The S-unit group U_S of L is the set of elements $\alpha \in L$ such that there is $\vec{e} \in \mathbb{Z}^{|S|}$ satisfying $\alpha\mathcal{O}_L = \prod_{\mathfrak{p} \in S} \mathfrak{p}^{e_\mathfrak{p}}$ where $\mathcal{O}_L$ is the maximal order of L. The computation of the S-unit group is a fundamental problem in computational number theory with many applications.

In this paper, we present an original algorithm for the computation of certain S-unit groups in real multiquadratic fields. The main motivation for the development of this algorithm is the computation of the ideal class group of $\mathcal{O}_L$. The computation of $\text{Cl}(\mathcal{O}_L)$ can be trivially deduced from the knowledge of an S-unit group where the classes of the elements of S generate $\text{Cl}(\mathcal{O}_L)$. The computation of the ideal class group is one of the four major tasks in computational number theory postulated by Zassenhaus [23, p. 2] (together with the computation of the unit group, the Galois group, and the ring of integers). In 1968, Shanks [26; 27] proposed an algorithm relying on the baby-step giant-step method to compute the class number and the regulator of a quadratic number field in time $O(|\Delta|^{1/4+\epsilon})$, or $O(|\Delta|^{1/5+\epsilon})$ under the extended Riemann hypothesis [21]. Then, a subexponential strategy for the computation of the

This work was supported by the U.S. National Science Foundation under grant 1839805, by the National Institute of Standards and Technology under grant 60NANB17D184, and by the Simons Foundation under grant 430128.
MSC2010: primary 11R04, 11R29, 11R65, 11Y99; secondary 11R11, 11R16, 11S20, 11Y50.
Keywords: ideal class group, S-unit group, multiquadratic fields.

group structure of the class group of an imaginary quadratic field was described in 1989 by Hafner and McCurley [20]. The expected running time of this method is

$$L_\Delta(1/2, \sqrt{2} + o(1)) = e^{(\sqrt{2}+o(1))\sqrt{\ln|\Delta| \ln\ln|\Delta|}}.$$

Buchmann [15] generalized this result to the case of infinite classes of number fields with fixed degree. Practical improvements to Buchmann's algorithm were presented in [18] by Cohen, Diaz y Diaz, and Olivier. Biasse [6] described an algorithm for computing the ideal class group and the unit group of $\mathcal{O} = \mathbb{Z}[\theta]$ in heuristic complexity bounded by $L_\Delta(1/3, c)$ for some $c > 0$ valid in certain classes of number fields. In [7; 10], Biasse and Fieker showed that there was a heuristic subexponential algorithm for the computation of the ideal class group in all classes of number fields. The methods of [10] can be specialized to the case of cyclotomic fields for a better asymptotic complexity [8]. The computation of the ideal class group is also the subject of study in the context of quantum computing. It was recently proved (under the GRH) by Biasse and Song that there is a quantum polynomial time algorithm for the computation of the ideal class group of an arbitrary field [13]. The most efficient practical implementations of algorithms for the computation of the ideal class group are either based on the quadratic sieve [12; 5; 4; 11] for quadratic fields or on the number field sieve [9] for number fields of higher degree.

The computation of S-units is also instrumental in the resolution of norm equations [28]. Indeed, it is the bottleneck of the resolution in x of $\mathcal{N}_{L/K}(x) = a$ for a given $a \in K$ where L/K is a Galois extension. This computational problem is closely related to Hilbert's 10th problem, for which there is no efficient *general* solution.

Contributions. Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a real multiquadratic number field. We define $d = \prod_{i \le n} d_i$ and $\Delta = \mathrm{disc}(L)$.

- We describe an algorithm to compute $\mathrm{Cl}(\mathcal{O}_L)$ in heuristic complexity $\mathrm{Poly}(\log(\Delta))e^{\widetilde{O}(\sqrt{\log d})}$.
- We describe a heuristic algorithm for the computation of the S-class group and the S-unit group of L in time $\mathrm{Poly}(\log(\Delta), \mathrm{Size}(S))e^{\widetilde{O}(\sqrt{\log d})}$.
- We report on the performance of an implementation of our algorithms.

Our recursive approach is based on the unit group computation of [3] which we extended to the more general problem of the computation of the S-unit group. In the case where d is small compared to Δ, our method for computing class groups, S-class groups, and S-unit groups runs in heuristic polynomial time in $\log(\Delta)$ (and in the size of S) where $\log(x)$ is the bit size of the integer x. This is ensured when $\log(d) \le \log(\log(\Delta))^c$ for some constant $c < 2$. For example, this is the case when the d_i are the first n consecutive primes. This is the first nonquantum algorithm that runs in polynomial time on infinite classes of number fields. The main ingredient of our recursion strategy is not restricted to multiquadratic fields. We can take advantage of computations in subfields whenever there are two different $\sigma, \tau \in \mathrm{Gal}(L/\mathbb{Q})$ of order two. General subfields might not enjoy the same general recursive structure as multiquadratic fields, but we expect that the reduction to the computation in subfields will improve the performance of class group algorithms. The application of these methods to more general fields was left for future work.

2. Preliminaries

2A. ***Number fields.*** A number field K is a finite extension of $\mathbb{Q}$. Its ring of integers $\mathcal{O}_K$ has the structure of a lattice of degree $n = [K : \mathbb{Q}]$. A number field has $r_1 \le n$ real embeddings $(\sigma_i)_{i\le r_1}$ and $2r_2$ complex embeddings $(\sigma_i)_{r_1<i\le 2r_2}$ (coming as r_2 pairs of conjugates). The pair (r_1, r_2) is the signature of K. The field K is isomorphic to $\mathcal{O}_K \otimes \mathbb{Q}$. The norm of an element $x \in K$ is defined by $\mathcal{N}(x) = \prod_i \sigma_i(x)$. Let $(\alpha_i)_{i\le n}$ such that $\mathcal{O}_K = \bigoplus_i \mathbb{Z}\alpha_i$; then the discriminant of K is $\Delta(K) := \det^2(T_2(\alpha_i, \alpha_j))$, where T_2 is defined by $T_2(x, x') := \sum_i \sigma_i(x)\overline{\sigma_i}(x')$. When there is no ambiguity, we simply denote it by Δ.

2B. ***Units of $\mathcal{O}_K$.*** Elements $u \in \mathcal{O}_K$ that are invertible in $\mathcal{O}_K$ are called units. Equivalently, they are the elements $u \in K$ such that $(u) := (u)\mathcal{O}_K = \mathcal{O}_K$. The unit group of $\mathcal{O}_K$ where K is a real multiquadratic field has rank $r = n-1$ and has the form $\mathcal{O}_K^* = \mu \times \langle\epsilon_1\rangle \times \cdots \times \langle\epsilon_r\rangle$ where μ are roots of unity (torsion units) and the ϵ_i are nontorsion units. Such $(\epsilon_i)_{i\le r}$ are called a system of fundamental units of $\mathcal{O}_K$. Units generate a lattice $\mathcal{L}$ of rank r in $\mathbb{R}^{r+1}$ via the embedding $x \in K \mapsto \mathrm{Log}(x) := (\ln(|\sigma_1(x)|), \ldots, \ln(|\sigma_{r+1}(x)|))$. The volume R of $\mathcal{L}$ is an invariant of K called the regulator. The regulator R and the class number h satisfy $hR = (|\mu|\sqrt{|\Delta|}/(2^{r_1}(2\pi)^{r_2}))\lim_{s\to 1}((s-1)\zeta_K(s))$, where $\zeta_K(s) = \sum_{\mathfrak{a}} 1/\mathcal{N}(\mathfrak{a})^s$ is the usual ζ-function associated to K and $|\mu|$ is the cardinality of μ the group of torsion units. This allows us to derive a bound h^* in polynomial time under GRH that satisfies $h^* \le hR < 2h^*$ [2].

2C. ***Multiquadratic fields.*** In this paper, we focus on towers of quadratic extensions.

Definition 2.1. Let $d_1, \ldots, d_n$ be squarefree integers that are multiplicatively independent modulo squares (i.e., they are independent in $\mathbb{Q}^\times/(\mathbb{Q}^\times)^2$). Then $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ is called a multiquadratic field and $N := [L : \mathbb{Q}] = 2^n$. Its Galois group $\mathrm{Gal}(L/\mathbb{Q}) := \{\text{Automorphisms of } L \text{ that fix } \mathbb{Q}\}$ is isomorphic to $(\mathbb{Z}/2\mathbb{Z})^n$.

When $n = 1$, the field $L = \mathbb{Q}(\sqrt{d_1})$ is simply called a quadratic field. In this paper, we focus on real multiquadratic fields, that is, those that satisfy $d_i > 0$ for all $i \le n$. The discriminant of a real multiquadratic field is given to us by an explicit formula. This is useful for the computation of its maximal order.

Lemma 2.2. *Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ a multiquadratic field as given above and $\prod_{j=1}^s p_j^{m_j}$ with $p_1 < p_2 < \cdots < p_s$ be the factorization of $\prod_{i=1}^n d_i$. Then $\Delta(L) = (2^a p_1 \cdot p_2 \cdots p_s)^{2^{n-1}}$ where*

$$a = \begin{cases} 0 & \text{if } d_i \equiv 1 \bmod 4 \text{ (for all } 1 \le i \le n), \\ 2 & \text{if } p_1 = 2 \text{ and } p_i \equiv 1 \bmod 4 \text{ (for all } 2 \le i \le n), \\ & \qquad \text{or } p_1 \ne 2 \text{ and there exists } i \text{ such that } p_i \equiv 3 \bmod 4, \\ 3 & \text{otherwise.} \end{cases}$$

Proof. This follows from Theorem 2.1 of [25]. □

If we take $d_1, d_2, \ldots, d_n$ to be the first n primes, then their product is the primorial $p_n\# \approx e^{(1+o(1))n\log n}$. Combining this with Lemma 2.2 gives $\ln \Delta(L) \approx \frac{1}{2}Nn\log n = \frac{1}{2}N\log N\log\log N$.

2D. ***Class groups.*** Elements of the form $\mathfrak{I}/d$ where $\mathfrak{I} \subseteq \mathcal{O}_K$ is an ideal of the ring of integers of K and $d > 0$ are called fractional ideals. Ideals of $\mathcal{O}_K$ are also referred to as integral ideals. Fractional ideals have the structure of a $\mathbb{Z}$-lattice of degree $n = [K : \mathbb{Q}]$, and they form a multiplicative group $\mathcal{I}$. Elements of $\mathcal{I}$ admit a unique decomposition as a product of nonzero prime ideals of $\mathcal{O}_K$ (with possibly negative exponents). The norm of integral ideals is given by $\mathcal{N}(\mathfrak{I}) := [\mathcal{O}_K : \mathfrak{I}]$, which extends to fractional ideals by $\mathcal{N}(\mathfrak{I}/\mathfrak{J}) := \mathcal{N}(\mathfrak{I})/\mathcal{N}(\mathfrak{J})$. The norm of a principal (fractional) ideal agrees with the norm of its generator $\mathcal{N}(x\mathcal{O}_K) = |\mathcal{N}(x)|$. The principal fractional ideals $\mathcal{P}$ of K are a subgroup of $\mathcal{I}$ and the ideal class group of $\mathcal{O}_K$ is defined by $\mathrm{Cl}(\mathcal{O}_K) := \mathcal{I}/\mathcal{P}$. We denote by $[\mathfrak{a}]$ the class of a fractional $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O}_K)$ and by h the cardinality of $\mathrm{Cl}(\mathcal{O}_K)$ which is a finite group. Let $\mathfrak{a}$, $\mathfrak{b}$ be two fractional ideals of K. We have $[\mathfrak{a}] = [\mathfrak{b}]$ if and only if there is $\alpha \in K$ such that $\mathfrak{a} = (\alpha)\mathfrak{b}$. We also denote this property by $\mathfrak{a} \sim \mathfrak{b}$.

2E. ***How to compute class groups.*** The best asymptotic algorithms to compute the ideal class group of $\mathcal{O}_K$ follow the general framework deriving from the algorithm of Hafner and McCurley [20] (subsequently generalized by Buchmann [15] and Biasse and Fieker [10]). Let $B > 0$ be a bound and define a factor base as $\mathcal{B} := \{\text{nonzero prime ideals } \mathfrak{p} \text{ with } \mathcal{N}(\mathfrak{p}) \leq B\}$. We refer to B as the *smoothness bound*. We compute a generating set of the lattice Λ of all the vectors $(e_1, \ldots, e_m) \in \mathbb{Z}^m$ with $m := |\mathcal{B}|$ such that there exists $\alpha \in K$ with $(\alpha) = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_m^{e_m}$.

Definition 2.3 (relations). Let $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ be a set of nonzero prime ideals of K. For each S-unit $\alpha \in K$ with $\vec{e} = (e_1, \ldots, e_s)$ such that $(\alpha) = \prod_i \mathfrak{p}_i^{e_i}$, we define the *relation* associated with α by $\mathcal{R}_{S,K}(\alpha) := (\alpha, \vec{e})$. The relations of K for the set S form a group denoted by $\mathcal{R}el_S(K)$.

When $B > 12\ln^2|\Delta|$, the classes of ideals in $\mathcal{B}$ generate $\mathrm{Cl}(\mathcal{O}_K)$ under the GRH [1, Theorem 4]. Therefore, $(\mathcal{B}, \Lambda)$ is a presentation of the group $\mathrm{Cl}(\mathcal{O}_K)$ and the search for a generating set of the relations $\mathcal{R}el_S(K)$ for $S = \mathcal{B}$ is equivalent to computing the group structure of $\mathrm{Cl}(\mathcal{O}_K)$. Indeed, the morphism

$$\begin{array}{rcccc} \mathbb{Z}^m & \xrightarrow{\varphi} & \mathcal{I} & \xrightarrow{\pi} & \mathrm{Cl}(\mathcal{O}_K), \\ (e_1, \ldots, e_m) & \longrightarrow & \prod_i \mathfrak{p}_i^{e_i} & \longrightarrow & \prod_i [\mathfrak{p}_i]^{e_i} \end{array}$$

is surjective, and the class group $\mathrm{Cl}(\mathcal{O}_K)$ is isomorphic to $\mathbb{Z}^m/\ker(\pi \circ \varphi) = \mathbb{Z}^m/\Lambda$.

2F. ***S-class groups and S-unit groups.*** Let $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ be a finite set of prime ideals of the number field K. We say that $x \in K$ is an S-integer if $v_\mathfrak{p}(x) \geq 0$ for all $\mathfrak{p} \notin S$. The set of S-integers is a ring denoted by $\mathcal{O}_{K,S}$. We define the S-unit group $U_{K,S}$ (or U_S if the field of definition is understood) as the elements $x \in K$ such that $v_\mathfrak{p}(x) = 0$ for all $\mathfrak{p} \notin S$. The group of S-units is finitely generated: $U_S = \mu(K) \times \langle \eta_1 \rangle \times \cdots \times \langle \eta_{r+s} \rangle$ where $\mu(K)$ is the set of the roots of unity of K, and $\eta_1, \ldots, \eta_{s+r}$ are torsion-free generators. The rank of its torsion-free part equals $r + s$ where r is the rank of the torsion-free part of the unit group U_K. Let $\mathcal{I}_S$ be the group of fractional ideals of $\mathcal{O}_{K,S}$, and $\mathcal{P}_S$ its subgroup of principal ideals. We define the S-class group by $\mathrm{Cl}_S(\mathcal{O}_{K,S}) = \mathcal{I}_S/\mathcal{P}_S$.

3. S-units of quadratic fields

In this section, we assume that $L = \mathbb{Q}(\sqrt{d})$ for $d > 0$ a squarefree integer. The calculation of the S-unit group for S a set of prime ideals of L is done by using the approach of Simon [28, §I.1.2]. Together with the subexponential strategy for computing the ideal class group derived from the Hafner–McCurley algorithm [20], the S-unit group of L can be computed in time $\mathrm{Poly}(\mathrm{Size}(S)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$. These algorithms have been extensively studied, in particular in [20; 15; 10; 28]. Therefore, we only give a brief sketch of the algorithm and will focus on the run time and the format of the output.

3A. ***Computing the class group.*** First, let $B \in e^{\widetilde{O}(\sqrt{\log d})}$ be a large enough smoothness bound such that the nonzero prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ of L with norm less than B generate $\mathrm{Cl}(\mathcal{O}_L)$. Note that $k \in e^{\widetilde{O}(\sqrt{\log d})}$. The computation of $\mathrm{Cl}(\mathcal{O}_L)$ starts with the collection of $\delta_1, \ldots, \delta_l$ for some $l \in \widetilde{O}(k)$ such that for all $i \le l$ there exist $(a_{i,1}, \ldots, a_{i,k})$ with $(\delta_i) = \prod_j \mathfrak{p}_j^{a_{i,j}}$. The δ_i and the $a_{i,j}$ are all polynomial size in $\log(d)$. Then there are unimodular matrices $U \in \mathrm{GL}_l(\mathbb{Z})$ and $V \in \mathrm{GL}_k(\mathbb{Z})$ such that

$$\mathrm{SNF}(A) = UAV = \left(\begin{array}{ccc} d_1 & & (0) \\ & \ddots & \\ (0) & & d_k \\ \hline & (0) & \end{array}\right),$$

where $\mathrm{SNF}(A)$ denotes the Smith normal form of A. The unimodular matrices U, V can be found in polynomial time [29] (in the dimension and the bit size of the entries of A), and their entries have polynomial size in the dimension of A and the bit size of its coefficients. This means that $\log(|U|), \log(|V|) \in e^{\widetilde{O}(\sqrt{\log d})}$ where $|U|$ denotes a bound on the absolute values of the entries of U. Let $\mathcal{L} \subseteq \mathbb{Z}^k$ be the lattice generated by the rows of A. Then

$$\mathrm{Cl}(\mathcal{O}_L) \simeq \mathbb{Z}^k/\mathcal{L} \simeq \mathbb{Z}/d_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/d_k\mathbb{Z}.$$

Let $\mathfrak{g}_j := \prod_{i \le k} \mathfrak{p}_i^{v_{i,j}}$; we have $\mathrm{Cl}(\mathcal{O}_L) \simeq \langle [\mathfrak{g}_1] \rangle \times \cdots \times \langle [\mathfrak{g}_k] \rangle$. In addition, let $\beta_i := \prod_{j \le l} \delta_j^{u_{i,j}}$, for $i \le k$. We do not evaluate this product. We have $\mathfrak{g}_i^{d_i} = (\beta_i)$. Overall, the complexity of this calculation is in $e^{\widetilde{O}(\sqrt{\log d})}$.

3B. ***Computing the S-unit group.*** Let S be a set of primes $\mathfrak{q}_1, \ldots, \mathfrak{q}_s$ of L. To get the S-class group and the S-unit group we add extra relations to $\mathcal{L}$. More specifically, we need to identify the classes of $\mathrm{Cl}(\mathcal{O}_L)$ that are represented by a product of primes in S with the trivial class of $\mathrm{Cl}_S(\mathcal{O}_{L,S})$. The ideal class of each of the elements of S can be represented as a product of the classes of the $\mathfrak{g}_i$. In time $e^{\widetilde{O}(\sqrt{\log d})}$ (and polynomial in $\log(\mathcal{N}(\mathfrak{q}_i))$), one can find polynomial size $x_1, \ldots, x_k$ and $\beta_{i+k} \in L$ such that $\mathfrak{q}_i = (\beta_{i+k}) \prod_j \mathfrak{p}_j^{x_j}$ with standard methods derived from [20]. Then for each j, $\mathfrak{p}_j = \prod_{i \le k} \mathfrak{g}_i^{v'_{i,j}}$ where the $v'_{i,j}$ are the coefficients of V^{-1}, we readily find vectors $\vec{e}_i \in \mathbb{Z}^k$ with entries having polynomial size in k (that is in $e^{\widetilde{O}(\sqrt{\log d})}$) such that $\mathfrak{q}_i = (\beta_{i+k}) \prod_{j \le k} \mathfrak{g}_j^{e_{i,j}}$. The vectors $\vec{e}_i$ are precisely the new additions

needed to expand $\mathcal{L}$. We get a new relation matrix

$$B = \left(\begin{array}{ccc} d_1 & & (0) \\ & \ddots & \\ (0) & & d_k \\ \hline e_{1,1} & \dots & e_{1,k} \\ \vdots & & \vdots \\ e_{s,1} & \dots & e_{s,k} \end{array}\right).$$

As for the computation of $\mathrm{Cl}(\mathcal{O}_L)$, the SNF of B gives the elementary divisors of the cyclic decomposition of $\mathrm{Cl}_S(\mathcal{O}_{K,S})$. Meanwhile, let $\vec{w}_1, \dots, \vec{w}_{1+s}$ be a basis for the left kernel of B (in general the dimension is $r+s$ where r is the rank of the unit group of L). This kernel is found in polynomial time in the dimension of B and the size of its entries, that is in time $\mathrm{Poly}(s) \cdot e^{\widetilde{O}(\sqrt{\log d})}$. The entries of the kernel vectors have size in $\mathrm{Poly}(s) \cdot e^{\widetilde{O}(\sqrt{\log d})}$, and $U_S = \mu \times \langle \gamma_1 \rangle \times \cdots \times \langle \gamma_{1+s} \rangle$ where $\mu = \{\pm 1\}$ are the torsion units of $\mathcal{O}_L$ and $\gamma_i := \prod_{j \le k+s} \delta_j^{w_{i,j}}$.

Proposition 3.1. *Let $d > 0$ be a squarefree integer, $L = \mathbb{Q}(\sqrt{d})$, and S be a set of prime ideals of L with $|S| = s$. Then the S-unit group algorithm of* [28, §I.1.2] *returns $l \in e^{\widetilde{O}(\sqrt{\log d})}$ polynomial size elements $\delta_i \in L$ and $s+1$ vectors $\vec{c}_i$ with entries of size in $\mathrm{Poly}(s) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ such that the $s+1$ elements $\gamma_i := \prod_{j \le l} \delta_j^{c_{i,j}}$ generate the S-unit group of L. The overall complexity of this procedure is in $\mathrm{Poly}(\mathrm{Size}(S)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ where the size of S is in $O(s \cdot \max_{\mathfrak{p} \in S} \log(\mathcal{N}(\mathfrak{p})))$.*

4. Recursive computation of S-units

Let S be a set of nonzero prime ideals in L that is invariant under the action of $\mathrm{Gal}(L/\mathbb{Q})$ (that is, for all $\mathfrak{p} \in S$ and all $\sigma \in \mathrm{Gal}(L/\mathbb{Q})$, $\mathfrak{p}^\sigma \in S$). In this section, we introduce a recursive method for finding a generating set of $\mathcal{R}el_S(L)$ which is the group of elements of the form $\mathcal{R}_{S,L}(\alpha) = (\alpha, \vec{e})$ such that $(\alpha) = \prod_{\mathfrak{p}_i \in S} \mathfrak{p}_i^{e_i}$. Our strategy consists of deriving the S-unit group in L from that of three subfields of L. When we reach the leaves of this recursion tree, we use the methods of Section 3 for computing the S-unit group directly on the quadratic field.

4A. ***High-level description of the algorithms.*** Let L be a multiquadratic number field and let σ, τ be two distinct nontrivial automorphisms of L. Let $\sigma\tau := \sigma \circ \tau$ and K_ℓ be the subfield of L fixed by $\ell \in \{\sigma, \tau, \sigma\tau\}$. Let S be a set of prime ideals of the ring of integers $\mathcal{O}_L$ with L stable by the action of $\mathrm{Gal}(L/\mathbb{Q})$, and for each $\ell \in \{\sigma, \tau, \sigma\tau\}$ let us define $S_\ell := \{\mathfrak{p} \cap K_\ell \mid \mathfrak{p} \in S\}$. We recover a generating set of $\mathcal{R}el_S(L)$ from generating sets of $\mathcal{R}el_{S_\sigma}(K_\sigma)$, $\mathcal{R}el_{S_\tau}(K_\tau)$, and $\sigma(\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau}))$. Our result follows from two crucial observations.

(1) The subgroup U of $\mathcal{R}el_S(L)$ generated by the lifts of $\mathcal{R}el_{S_\sigma}(K_\sigma)$, $\mathcal{R}el_{S_\tau}(K_\tau)$, and $\sigma(\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau}))$ contains all the squares of relations in $\mathcal{R}el_S(L)$.

(2) There is an algorithm that efficiently produces elements of U that are square of relations in $\mathcal{R}el_S(L)$, and then computes their square root.

Algorithm 1: High-level description of recursive S-unit computation of L

Input: Real multiquadratic field L, ring of integers $\mathcal{O}_L$ of L, set of primes S of $\mathcal{O}_L$ stable under the action of $\mathrm{Gal}(L/\mathbb{Q})$.
Result: A basis for $\mathcal{R}el_S(L)$.

```
if [L : Q] = 2 then
    Use the method of [28, §I.1.2] to compute a basis Λ of Rel_S(L).
    return Λ
σ, τ ← distinct nonidentity automorphisms of L.
for ℓ ∈ {σ, τ, στ} do
    K_ℓ ← fixed field of ℓ.
    Λ_ℓ ← basis of Rel_{S_ℓ}(K_ℓ).
Λ ← Λ_σ ∪ Λ_τ ∪ σ(Λ_στ).
Find a basis Λ_2 of the lattice of relations generated by Λ that are squares.
Λ_2 ← square roots of the elements in Λ_2.
Λ ← basis of the lattice generated by Λ ∪ Λ_2.
return Λ
```

When the recursive tree reaches a quadratic subfield K_ℓ of L, it uses the subexponential algorithm of Simon [28, §I.1.2] to return the S_ℓ-unit group. The high-level description of this strategy is summarized in Algorithm 1. Note that the ring of integers $\mathcal{O}_L$ is part of the input. In general, the computation of $\mathcal{O}_L$ is as hard as the factorization of the discriminant of L, but in the particular case of multiquadratic fields, there is an efficient algorithm for this task [17].

4B. ***Lifting relations.*** To compute $\mathcal{R}el_S(L)$, we use the relations from $\mathcal{R}el_{S_\sigma}(K_\sigma)$, $\mathcal{R}el_{S_\tau}(K_\tau)$, and $\sigma(\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau}))$ where $\sigma, \tau \in \mathrm{Gal}(L/\mathbb{Q})$ and S_ℓ, K_ℓ are defined in Section 4A. Therefore, given relations in a subfield K_σ of L, we need to be able to efficiently compute the corresponding relations in L.

Theorem 4.1. *Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a multiquadratic field. Let K_σ be the (multi)quadratic subfield of L fixed by $\sigma \in \mathrm{Gal}(L/\mathbb{Q})$, $S_\sigma = \{\mathfrak{p}_i\}_{i \le s}$ where $\mathfrak{p}_i$ are prime ideals of K_σ, and $S = \{\mathfrak{P}_k \subset L \mid$ there exists $i \le s$ such that $\mathfrak{P}_k \cap K_\sigma = \mathfrak{p}_i\}$. Let $\mathcal{R}_{S_\sigma, K_\sigma}(\alpha) = (\alpha, \vec{e})$ be a relation in $\mathcal{R}el_{S_\sigma}(K_\sigma)$. Then $(\alpha, \vec{e}_L) := \mathcal{R}_{S,L}(\alpha) \in \mathcal{R}el_S(L)$ with $\vec{e}_L = (e_1 \vec{f}_1 \mid e_2 \vec{f}_2 \mid \cdots \mid e_s \vec{f}_s)$, where $\vec{f}_i$ satisfy $\mathfrak{p}_i \mathcal{O}_L = \prod_{j \le g_i} \mathfrak{P}_{k_{i,j}}^{f_{i,j}}$.*

Proof. Let $\alpha \in K_\sigma$ such that $(\alpha) = \prod_{\mathfrak{p}_i \in S} \mathfrak{p}_i^{e_i}$. Each prime ideal $\mathfrak{p}_i \in K_\sigma$ factors as $\mathfrak{p}_i \mathcal{O}_L = \prod_{j \le g_i} \mathfrak{P}_{k_{i,j}}^{f_{i,j}}$, where the $\mathfrak{P}_{k_{i,j}}$ are the prime ideals of L such that $\mathfrak{P}_{k_{i,j}} \cap K_\sigma = \mathfrak{p}_i$ and the $f_{i,j}$ are the corresponding ramification indices. Therefore, we have

$$(\alpha)\mathcal{O}_L = \prod_{\mathfrak{p}_i \in S} \mathfrak{p}_i^{e_i} \mathcal{O}_L = \prod_{\mathfrak{p}_i \in S} \prod_{j \le g_i} \mathfrak{P}_{k_{i,j}}^{e_i f_{i,j}}.$$

Thus, $(\alpha, (e_1 \vec{f}_1 \mid e_2 \vec{f}_2 \mid \cdots \mid e_s \vec{f}_s))$ is the relation corresponding to α in $\mathcal{R}el_S(L)$. □

Given the straightforward correspondence between $\mathcal{R}_{S_\sigma,K_\sigma}(\alpha) \in \mathcal{R}el_S(K_\sigma)$ and its lift in $\mathcal{R}el_S(L)$, we identify these two elements. The set $\mathcal{R}el_S(L)$ is also equipped with a natural group structure given by $(\alpha_1, \vec{e}_1) + (\alpha_2, \vec{e}_2) := (\alpha_1 \cdot \alpha_2, \vec{e}_1 + \vec{e}_2)$. We define the index of a subgroup U of $\mathcal{R}el_S(L)$ as that of the subgroup of U_S of the α such that there exists $\vec{e}$ with $(\alpha, \vec{e}) \in U$.

Lemma 4.2. *Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a multiquadratic field and let S be a set of prime ideals of L that is invariant under the action of* $\mathrm{Gal}(L/\mathbb{Q})$. *Let $\sigma, \tau \in \mathrm{Gal}(L/\mathbb{Q})$ be two different nonidentity isomorphisms, and define S_ℓ, K_ℓ of $\ell \in \{\sigma, \tau, \sigma\tau\}$ as in Section 4A. Let U be the group generated by $\mathcal{R}el_{S_\sigma}(K_\sigma) \cup \mathcal{R}el_{S_\tau}(K_\tau) \cup \sigma(\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau}))$ where*

$$\sigma(\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau})) := \{\mathcal{R}_{S_\tau,K_\tau}(\sigma(\alpha)) \mid \text{there exists } \vec{e} \text{ such that } (\alpha, \vec{e}) \in \mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau})\}.$$

Then $(\mathcal{R}el_S(L))^2 \subseteq U \subseteq \mathcal{R}el_S(L)$, where $(\mathcal{R}el_S(L))^2$ denotes the relations of the form $(\alpha^2, 2\vec{e})$ where $(\alpha, \vec{e}) \in \mathcal{R}el_S(L)$.

Proof. From Theorem 4.1, we know that the relations in $\mathcal{R}el_{S_\sigma}(K_\sigma)$, $\mathcal{R}el_{S_\tau}(K_\tau)$, and $\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau})$ lift naturally to relations in $\mathcal{R}el_S(L)$. Moreover, σ maps elements of $K_{\sigma\tau}$ to K_τ, and since S is invariant under the action of σ, a relation is mapped to another relation (modulo a permutation of the coefficients of the exponent vector). So the action of σ on $\mathcal{R}el_{S_{\sigma\tau}}(K_{\sigma\tau})$ is well defined, and $U \subseteq \mathcal{R}el_S(L)$.

For the other inclusion, let $(\alpha, \vec{e}) \in \mathcal{R}el_S(L)$. For each $\ell \in \{\sigma, \tau, \sigma\tau\}$, $\alpha \cdot \ell(\alpha)$ decomposes as a product of ideals in S_ℓ. Therefore, there are vectors $\vec{e}_\ell$ such that for each ℓ, $(\alpha \cdot \ell(\alpha), \vec{e}_\ell) \in \mathcal{R}el_{S_\ell}(K_\ell)$. Moreover,

$$\frac{\mathcal{N}_{L:K_\sigma}(\alpha)\mathcal{N}_{L:K_\tau}(\alpha)}{\sigma(\mathcal{N}_{L:K_{\sigma\tau}}(\alpha))} = \frac{\alpha \cdot \sigma(\alpha) \cdot \alpha \cdot \tau(\alpha)}{\sigma(\alpha \cdot \sigma\tau(\alpha))} = \alpha^2;$$

hence, $(\alpha^2, 2\vec{e}) = (\sigma(\alpha), \vec{e}_\sigma) + (\tau(\alpha), \vec{e}_\tau) - \sigma((\sigma\tau)(\alpha), \vec{e}_{\sigma\tau})$ is a linear combination of relations in $\mathcal{R}el(K_\sigma)$, $\mathcal{R}el(K_\tau)$ and $\sigma(\mathcal{R}el(K_{\sigma\tau}))$, so $(\mathcal{R}el(L))^2 \subseteq U$. □

4C. ***Representation of elements and square roots.*** The lift U of the relations in three different subfields yields a set of relations containing all the squares of the relations in $\mathcal{R}el_S(L)$. We need to solve two tasks:

(1) identification of a generating set of the squares of U and,

(2) for each square $(\alpha^2, 2\vec{e})$ found in (1), computation of $(\alpha, \vec{e})$.

p-th roots with saturation. Let us identify $U \subseteq \mathcal{R}el_S(L)$ with the elements $\alpha \in U_S$ such that there exists $\vec{e}$, $(\alpha, \vec{e}) \in U$. Let $b > 0$ such that $(U_S : U) = b$. For any prime $p \mid b$ there is some $\alpha \in U_S \setminus U$ such that $\alpha^p \in U$. The saturation technique of Biasse and Fieker [9] can be used to find elements in U_S that are not in U. Let us fix the prime p. For any residue degree 1 prime ideal $\mathfrak{Q} \notin S$ with $Q := \mathcal{N}(\mathfrak{Q})$ such that $p \mid Q - 1$ we define the map $\phi_\mathfrak{Q} : U \to \mathbb{F}_Q^*/(\mathbb{F}_Q^*)^p$ mapping S-units into the multiplicative group of the residue class field $\mathbb{F}_Q := \mathcal{O}_L/\mathfrak{Q}$ modulo p-th powers. The Chebotarev theorem [31] guarantees that if $\alpha \in U$ is not a p-th power, there will be some $\mathfrak{Q}$ such that $\phi_\mathfrak{Q}(\alpha)$ is nontrivial, i.e., α is not a p-th power modulo Q. To find p-th powers, we now simply intersect $\ker \phi_\mathfrak{Q}$ for sufficiently many $\mathfrak{Q}$. The elements $\alpha \in U/\left(\bigcap \ker \phi_\mathfrak{Q}\right)$ will have a p-th root in U_S but not in U. Suppose $(\alpha, \vec{e}) \in U$ with $\alpha \in U/\left(\bigcap \ker \phi_\mathfrak{Q}\right)$; then $(\sqrt[p]{\alpha}, \vec{e}/p)$ is a new relation that reduces the index of the lattice of currently found relations in $\mathcal{R}el_S(L)$.

Using quadratic characters for $p = 2$. When looking for square roots, we can use quadratic characters to find elements in elements $\alpha \in U/\left(\bigcap \ker \phi_{\mathfrak{Q}}\right)$ by following the approach of [3]. More specifically, in [3, §4.1], the map

$$\phi_{\mathfrak{Q}} : \mathbb{Z}[x_1, \ldots, x_n]/(x_1^2 - d_1, \ldots, x_n^2 - d_n) \simeq \mathbb{Z}[\sqrt{d_1}, \ldots, \sqrt{d_n}] \to \mathbb{F}_Q,$$

where $\mathfrak{Q}$ is a residue degree 1 prime ideal and $Q = \mathcal{N}(\mathfrak{Q})$, is defined by $x_i \mapsto s_i$ where s_i is a square root of d_i modulo Q. Elements of U_S have nonnegative valuation at $\mathfrak{Q}$ since it satisfies $\mathfrak{Q} \notin S$. We can use the characters defined in [3, §4.1] by $\chi_{\mathfrak{Q}}(\alpha) := (\phi_{\mathfrak{Q}}(\alpha)/Q) \in \{-1, 0, 1\}$. When α is a square, we have $\chi_{\mathfrak{Q}}(\alpha) = 1$. To find squares, we find the $\alpha \in U$ such that $\chi_{\mathfrak{Q}_i}(\alpha) = 1$ for $i \leq m$ where m is large enough. This boils down to the search for a kernel element of the linear map

$$\begin{aligned} U_S &\xrightarrow{X} \mathbb{Z}/2\mathbb{Z} \times \cdots \times \mathbb{Z}/2\mathbb{Z}, \\ \alpha &\longrightarrow (\log_{-1}(\chi_{\mathfrak{Q}_1}(\alpha)), \ldots, \log_{-1}(\chi_{\mathfrak{Q}_m}(\alpha))), \end{aligned}$$

where for each $x \in \{-1, 1\}$, $\log_{-1}(x)$ denotes the discrete logarithm of x in base -1. If α is a square, then necessarily $X(\alpha) = (0, \ldots, 0)$. On the other hand, if $X(\alpha) = (0, \ldots, 0)$, there is a nonzero probability that α might not be a square. Given generators $\alpha_1, \ldots, \alpha_k$ of U, we can find a generating set of the squares of elements of U_S. This contains the squares of elements $\alpha_{k+1}, \ldots, \alpha_{k+l}$ of U_S such that $\alpha_1, \ldots, \alpha_{k+l}$ generate U_S. We obtain these squares by finding the kernel of the matrix $A = (X(\alpha_i)) \in \mathbb{Z}^{k \times m}$.

Representation of the elements. We compute S-units in the quadratic fields by directly applying the subexponential algorithm of [28, §I.1.2]. As we saw in Section 3, the output of the computation in each quadratic field $K_l := \mathbb{Q}(\sqrt{d_{K_l}})$ for $l \leq 2^n := N$ is a set of $s+1$ elements γ_i that are represented by vectors of exponents $\vec{e}_i$ and k elements α_j such that $\gamma_i = \prod_{j \leq k} \delta_j^{e_{i,j}}$. The δ_j have polynomial size in $\log(d_{K_l})$, while $k \in e^{\widetilde{O}(\sqrt{\log(d_{K_l})})}$ and the entries of $\vec{e}_i$ have size in $\mathrm{Poly}(s) \cdot e^{\widetilde{O}(\sqrt{\log(d_l)})}$. In our algorithm these products are never evaluated in L. Indeed, the representation of such elements on the integral basis has exponential size, thus making any calculation on them prohibitively expensive.

To avoid this issue, we use the so-called compact representation described by Thiel [30]. Given $\eta \in K$, we can find polynomial size (in the logarithm of the field discriminant) elements $\eta_0, \eta_1, \ldots, \eta_v$ such that $\eta = \eta_0 \eta_1^2 \cdots \eta_v^{2^v}$. Given an element η in compact representation, we can easily perform the operations

- compute $X(\eta) = X(\eta_0)$,
- compute $\sqrt{\eta} = \pm\sqrt{\eta_0} \cdot \eta_1 \cdot \eta_2^2 \cdots \eta_v^{2^{v-1}}$, and
- compute $\sigma(\eta) = \sigma(\eta_0)\sigma(\eta_1)^2 \cdots \sigma(\eta_v)^{2^v}$ for $\sigma \in \mathrm{Gal}(L/\mathbb{Q})$.

The original compact representation of Thiel [30] can be adapted to run in polynomial time with respect to the input. In particular, if η is given as a product $\eta = {\eta'_1}^{e_1} \cdots {\eta'_{v'}}^{e_{v'}}$, then we can find a compact representation of η in polynomial time in $\max_i(\mathrm{Size}(\eta'_i))$, $\max_i \log(e_i)$, and v' [10, §5; 19, §4.4]. We compute the compact representation of the γ_i at the beginning of the recursion, and after each subsequent operation.

Algorithm 2: SUnitGivenSubgroup$(K, \alpha_1, \ldots, \alpha_k)$

Input: Real multiquadratic field $K \subseteq \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$, $\alpha_1, \ldots, \alpha_k$ such that $U_{K,S}^2 \subseteq \langle \alpha_1, \ldots, \alpha_k \rangle$.
Result: Generators of $U_{K,S}/\{\pm 1\}$.

1 $\chi_1, \ldots, \chi_m \leftarrow$ characters defined by $\mathfrak{Q}_i$ for $i \leq m$.
2 $A \leftarrow [\log_{-1}(\chi_i(\alpha_j))]_{i \leq m, j \leq k} \in \mathbb{F}_2^{m \times k}$.
3 $V \leftarrow$ basis of the left kernel of A.
4 **for** $i = 1, \ldots, \#V$ **do**
5 $\quad v_i \leftarrow \prod_j \alpha_j^{V_{ij}}$.
6 $\quad \beta_i \leftarrow \sqrt{v_i}$.
7 **return** $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_{\#V}$

By linearity, one can evaluate the characters $X(\gamma_i) = \sum_{j \leq k} e_{i,j} X(\delta_j)$ in time $k \cdot \mathrm{Poly}(\max_{i,j} \mathrm{Size}(e_{i,j})) \cdot \mathrm{Poly}(\max_i \mathrm{Size}(X(\delta_j)))$. As $\mathrm{Size}(X(\delta_j))$ is bounded by $m \cdot \max_i \log(\mathcal{N}(\mathfrak{Q}_i))$, the resulting complexity is in $\mathrm{Poly}(s, m, \log Q) \cdot e^{\widetilde{O}(\sqrt{\log(d)})}$ where $Q := \max_i \mathcal{N}(\mathfrak{Q}_i)$ and $d := \prod_{l \leq n} d_l$. Using the compact representation, the product of two elements, the image under a morphism $\sigma \in \mathrm{Gal}(L/\mathbb{Q})$, and the computation of the square root are straightforward operations with complexity in $\mathrm{Poly}(s, \log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log(d)})}$.

In the description of Algorithm 2, we identify field elements and their representation described above. As previously mentioned, all squares must map to elements of $\mathrm{LeftKernel}(A)$, but there is a chance that elements from $\mathrm{LeftKernel}(A)$ do not arise as the map of a square in K. In this case, the element s_i calculated in Step 5 is not a square, and the (formal) square root computed in Step 6 does not correspond to any element in K. The probability of success of Algorithm 2 is derived from a standard heuristic used for the computation of square roots in the number field sieve algorithm [16, §8]. This argument was also used for computing units of multiquadratic fields in [3, §4.2]. Let $U := \langle \alpha_1, \ldots, \alpha_k \rangle / \{\pm 1\}$. The rank of $U/(U \cap K^2)$ is at most $s + r$ where r is the rank of the unit group of K and $s := |S|$. Therefore, the dual $\mathrm{Hom}(U/(U \cap K^2), \mathbb{F}_2)$ is an $\mathbb{F}_2$ vector space of dimension at most $r + s$. Assuming that $\log_{-1} \chi_{\mathfrak{Q}_1}, \ldots, \log_{-1} \chi_{\mathfrak{Q}_m}$ are independent uniform random elements of this dual, they span the dual with probability at least $1 - 1/2^{m-r-s}$ by [16, Lemmma 8.2]. In that case, $X(\alpha) = 0$ implies $\alpha \in U \cap K^2$.

Heuristic 4.3. *Let K be a multiquadratic subfield of $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$, and let S be a set of prime ideals of K. Let $\alpha_1, \ldots, \alpha_k$ be elements generating $U_{K,S}^2$ and let $U := \langle \alpha_1, \ldots, \alpha_k \rangle / \{\pm 1\}$ Then morphisms of the form $\log_{-1} \chi_{\mathfrak{Q}_i}$ are uniformly distributed in $\mathrm{Hom}(U/(U \cap K^2), \mathbb{F}_2)$.*

Proposition 4.4. *Let K be a multiquadratic subfield of $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$, and let S be a set of prime ideals of K. Let $\alpha_1, \ldots, \alpha_k$ be elements generating $U_{K,S}^2$. Let r be the rank of the unit group of K and let $s := |S|$. Then the run time of Algorithm 2 is in $\mathrm{Poly}(s, m, \log(\Delta), \log Q) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ where m is the number of characters, $N = 2^n$, $Q = \max_{i \leq m} Q_i$, and $d = \prod_{i \leq n} d_i$. Algorithm 2 returns a generating set of $U_{K,S}$ with probability at least $1 - 1/2^{m-r-s}$ under Heuristic 4.3.*

Remark 4.5. The only subroutine that we have not formally analyzed is the creation of the $\chi_1, \ldots, \chi_m$. For that, we directly rely on the algorithm GoodPrime of [3]. It returns each prime in time $O(N)$. Thus, the calculation of $\chi_1, \ldots, \chi_m$ is in $O(mN)$.

4D. ***Overall procedure.*** We now have all the ingredients to specify the details of our recursive method to compute the S-unit group of $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ for a set of prime ideals S invariant under the action of the Galois group of L.

Theorem 4.6. *Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a real multiquadratic field of degree N and S be a set of prime ideals of L stable under* $\mathrm{Gal}(L/\mathbb{Q})$ *that does not contain any ideal above* 2*. Then under Heuristic 4.3, the elements $\beta_1, \ldots, \beta_{r+s}$ returned by Algorithm 3 generate the torsion-free part of U_S with probability $1 - 1/2^N$. The asymptotic complexity of Algorithm 3 is in* $\mathrm{Poly}(\mathrm{Size}(S), \log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ *where* $\mathrm{Size}(S) = s \cdot \max_{\mathfrak{p} \in S} \log(\mathcal{N}(\mathfrak{p}))$, $\Delta = \mathrm{disc}(L)$, *and* $d := \prod_{i \le n} d_i$.

Proof. Algorithm 3 is called $3^n \in \mathrm{Poly}(N)$ times. The run time of Algorithm 3 is essentially ruled by that of Algorithm 2 and by the cost of Steps 12 and 14. Moreover, the cost of the ideal arithmetic involved in the lifting of the relations is in $\mathrm{Poly}(\mathrm{Size}(S), \log(\Delta))$. The probability of success of the overall algorithm is at least $(1 - 1/2^{m-r-s})^N \sim 1 - N/2^{m-r-s}$ where r is the rank of the unit group of L. Therefore, a choice of $m \in \mathrm{Poly}(N, s)$ can ensure that the probability of success is at least $1 - 1/2^N$. With such

Algorithm 3: MQSUnits for S stable under $\mathrm{Gal}(L/\mathbb{Q})$

Input: Real multiquadratic field L, ring of integers $\mathcal{O}_L$ of L, and set of prime ideals S of $\mathcal{O}_L$ stable under $\mathrm{Gal}(L/\mathbb{Q})$.

Result: A basis of the relations $\mathcal{R}el_S(L)$.

1 $S_0 \leftarrow \{p_1, \ldots, p_s\}$ where for all $i \le s$, there exists $\mathfrak{p} \in S$ such that $p_i \mid \mathfrak{p}$.
2 **if** $[L : \mathbb{Q}] = 2$ **then**
3 $\quad \Lambda \leftarrow$ basis of $\mathcal{R}el_S(L)$ using [28, Algorithm I.1.2].
4 $\quad$ **return** Λ
5 $\sigma, \tau \leftarrow$ distinct nonidentity automorphisms of L.
6 **for** $\ell \in \{\sigma, \tau, \sigma\tau\}$ **do**
7 $\quad K_\ell \leftarrow$ fixed field of ℓ.
8 $\quad S \leftarrow \{\mathfrak{p} \subseteq K_\ell \mid$ there exists $e \in \mathbb{Z}$ and $p \in S_0$ such that $\mathcal{N}(\mathfrak{p}) = p^e\}$.
9 $\quad \Lambda_\ell \leftarrow \mathrm{MQSUnits}(K_\ell, S)$.
10 $\Lambda_U \leftarrow \Lambda_\sigma \cup \Lambda_\tau \cup \sigma(\Lambda_{\sigma\tau})$.
11 $\Lambda := \{(\alpha_1, \vec{e}_1), \ldots, (\alpha_k, \vec{e}_k)\} \leftarrow \mathrm{SUnitGivenSubgroup}(L, \Lambda_U)$ (Algorithm 2).
12 $A \leftarrow (\vec{e}_i)_{i \le k}$. Compute $U \in \mathrm{GL}_k(\mathbb{Z})$ such that $UA = \left(\frac{H}{(0)}\right)$ is the HNF of A.
13 For $i = 1, \ldots, s$: $\beta_i \leftarrow \prod_{j \le k} \alpha_j^{U_{i,j}}$.
14 Compute a basis $\vec{w}_1, \ldots, \vec{w}_r$ of the left kernel of A.
15 For $i = 1, \ldots, r$, $\beta_{s+i} \leftarrow \prod_{j \le k} \alpha_j^{w_{i,j}}$.
16 **return** $(\beta_1, \vec{H}_1), \ldots, (\beta_s, \vec{H}_s), (\beta_{s+1}, \vec{0}), \ldots, (\beta_{s+r}, \vec{0})$

a choice of m, we can also ensure that $Q \in \mathrm{Poly}(N, s)$. Finally, in Steps 12 and 14, the coefficients of U and of the $\vec{w}_i$ are in $\mathrm{Poly}(s, \log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$. This allows us to bound the run time of the field operations of Steps 13 and 15 (in compact representation). Moreover, the run time of Steps 12 and 14 is also in $\mathrm{Poly}(s, \log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$, which proves the statement. □

The result of Algorithm 3 can be certified in polynomial time under the generalized Riemann hypothesis if the prime ideals in S generate the ideal class group of L. This is the case in all the applications that are considered in Section 5, including the computation of arbitrary S-unit groups. The only way Algorithm 3 can fail is if Algorithm 2 identifies nonsquares as squares. If this is the case, then the set of relations returned by Algorithm 3 contains elements that are not in $\mathcal{R}el_S(L)$. Let $h_0 := \det(H)$ and R_0 be the volume of the lattice generated by $\mathrm{Log}(\beta_i)$ for $i = s+1, \ldots, s+r$. If the result is correct, then $h_0 = h$ the class number of $\mathcal{O}_L$ while $R_0 = R$ the regulator of L. If not, then $h_0 R_0 \leq \frac{1}{2} hR$ (i.e., $\mathcal{R}el_S(L)$ is a finite index subgroup of the output of Algorithm 3). An estimate for hR can be found in polynomial time under the GRH by using the methods of [2].

Proposition 4.7. *Under the GRH, the result of Algorithm 3 can be certified in polynomial time if S includes a generating set of the ideal class group of $\mathcal{O}_L$.*

5. Applications of the S-unit computation algorithm

The S-unit group computation of Section 4 can be used to compute ideal class groups, S-class groups, and (arbitrary) S-unit groups.

5A. ***Ideal class group computation.*** As explained in Section 2E, the computation of $\mathrm{Cl}(\mathcal{O}_L)$ can be done by searching for a basis of the relations between a generating set of the classes of $\mathrm{Cl}(\mathcal{O}_L)$. Once such a generating set is found, then the strategy is the same as in [20], which was sketched in Section 3.

Proposition 5.1. *Let $L = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_n})$ be a real multiquadratic field of degree N an discriminant Δ. Under the GRH, Algorithm 4 successfully returns the ideal class group of $\mathcal{O}_L$ with probability $1 - 1/2^N$ in time $\mathrm{Poly}(\log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ where $d = \prod_{i \leq n} d_i$. The result of Algorithm 4 can be certified in polynomial time in $\log(\Delta)$.*

Algorithm 4: Computation of $\mathrm{Cl}(\mathcal{O}_L)$

Input: Ring of integers $\mathcal{O}_L$ of a real multiquadratic field L of degree N and discriminant Δ.
Result: Class group of $\mathcal{O}_L$.

1 Compute $S := \{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leq 12 \ln^2(\Delta)\}$.
2 $(\alpha_1, \vec{H}_1), \ldots, (\alpha_s, \vec{H}_s), (\alpha_{s+1}, \vec{0}), \ldots, (\alpha_{s+r}, \vec{0}) \leftarrow$ output of Algorithm 3.
3 $\mathrm{diag}(d_1, \ldots, d_s) \leftarrow \mathrm{SNF}(H)$.
4 **return** $\mathbb{Z}/d_1\mathbb{Z} \times \ldots \times \mathbb{Z}/d_s\mathbb{Z}$

$[L:\mathbb{Q}]$	Algorithm 4	Magma	Sage	$\mathrm{Cl}(\mathcal{O}_L)$
8	81.3	1.18	0.02	trivial
16	455	14.4	0.87	$C_4 \times C_4$
32	3738	3971	68.9	$C_2 \times C_4 \times C_8^4$
64	$3.79 \cdot 10^4$		$> 8.5 \cdot 10^5$	$C_2^9 \times C_4^3 \times C_8 \times C_{16}^4 \times C_{48} \times C_{240}$
128	$5.42 \cdot 10^5$			$C_2^{10} \times C_4^{16} \times C_8^{13} \times C_{16}^2 \times C_{48}^6 \times C_{96}^3 \times C_{48} \times C_{960}$

Table 1. Comparison of class group routine run time.

Corollary 5.2. *When $d = \prod_{i \le n} d_i$ satisfies $\log(d) < \log(\log(\Delta))^c$ for some constant $c < 2$, then Algorithm 4 returns the ideal class group of $\mathcal{O}_L$ with probability $1 - 1/2^N$ in polynomial time in $\log(\Delta)$.*

We showcase the effect of our algorithm on classes of multiquadratic fields with small d_i by computing the class group of the degree 128 multiquadratic field $L = \mathbb{Q}(\sqrt{5}, \sqrt{13}, \sqrt{17}, \sqrt{29}, \sqrt{37}, \sqrt{41}, \sqrt{53})$ and its subfields

$$\mathbb{Q}(\sqrt{5}, \sqrt{13}, \sqrt{17}), \ldots, \mathbb{Q}(\sqrt{5}, \sqrt{13}, \sqrt{17}, \sqrt{29}, \sqrt{37}, \sqrt{41}).$$

We implemented Algorithm 4 and ran experiments on a single core of an Intel Xeon E5-2650 v3 2.30 GHz processor with 512 GB of RAM running version 7.5.1 of Sage [24]. For the low level multiquadratic arithmetic, we used the methods of [3]. For the Sage experiments the class_group(proof = False) method was used. Note that Sage's class group routine directly calls that of Pari/GP [22]. We also ran the class group routine of Magma V.2.24 on the same fields. Magma [14] works at a higher level of rigor by only returning results that are at least certified under GRH (we ran the command ClassGroup(K:Proof:="GRH")). Therefore, the comparison with Sage is not entirely relevant. In degree 64, the computation with Magma had to be terminated after 24 hours since it had exhausted the machine's memory.

Although slower for small degrees, our method is the only implementation that is able to compute the class group of multiquadratic fields of degree more than 32. We can see on Table 1 that the run time (in CPU seconds) of Algorithm 4 is consistent with a polynomial run time in $\log(\Delta)$. Our algorithm is parallelizable on several levels: subtrees of the recursion tree are independent, as well as computations modulo the $(Q_i)_{i \le m}$. Therefore, we anticipate that a parallel version of our algorithm could reach degrees 256 and 512.

5B. ***S-class group and S-unit group computation.*** Algorithm 3 computes the S-unit group with the restriction that S contains all conjugates of any $\mathfrak{p} \in S$ under the action of $\mathrm{Gal}(L/\mathbb{Q})$. As shown in Section 3, the S-class group boils down to the search for the lattice of relations between the generators $(\mathfrak{g}_i)_{i \le s_0}$ of $\mathrm{Cl}(\mathcal{O}_L)$ which we enlarge with new relations of the form $\mathfrak{q}_j \sim \prod_{i \le s_0} \mathfrak{g}_i^{x_{i,j}}$. The SNF of this enlarged relation lattice gives the elementary divisors of the S-class group while its kernel reveal the S-unit group.

Algorithm 5: S-class group and S-unit group computation

Input: Real multiquadratic field L of degree N, ring of integers $\mathcal{O}_L$ of L, and a set S of prime ideals of $\mathcal{O}_L$.
Result: S-unit group and S-class group of L.

Compute $S_0 := \{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leq 12\ln^2(\Delta)\}$ for $\Delta = \mathrm{disc}(L)$.
$S_0 \leftarrow S \cup \{\mathfrak{q}^\sigma \mid \mathfrak{q} \in S, \sigma \in \mathrm{Gal}(L/\mathbb{Q})\}$.
$(\alpha_1, \vec{H}_1), \ldots, (\alpha_{s_0}, \vec{H}_{s_0}), (\alpha_{s_0+1}, \vec{0}), \ldots, (\alpha_{s_0+r}, \vec{0}) \leftarrow$ output of Algorithm 3.
Compute U, V such that $U\left(\frac{H}{(0)}\right)V = \left(\frac{\mathrm{SNF}(H)}{(0)}\right)$ with $\mathrm{SNF}(H) = \mathrm{diag}(d_i)_{i\leq s_0}$.
For $j \leq s_0$, define $\mathfrak{g}_j := \prod_{i\leq s_0} \mathfrak{p}_i^{V_{i,j}}$ (here, $\mathrm{Cl}(\mathcal{O}_L) \simeq \bigoplus_{i\leq k}\langle[\mathfrak{g}_i]\rangle$).
$V' \leftarrow V^{-1}$.
For each $j \leq s$, find $j_0 \leq s_0$ such that $\mathfrak{q}_j = \mathfrak{p}_{j_0}$.
$\vec{x}_j \leftarrow (V'_{1,j_0}, \ldots, V'_{s_0,j_0})$ (here $\mathfrak{q}_j = \prod_{i\leq s_0} \mathfrak{g}_i^{x_{i,j_0}}$).
Let $M = \left(\frac{H}{(\vec{x}_i)_{i\leq s}}\right)$.
$\mathrm{diag}(d'_i)_{i\leq s_0} \leftarrow \mathrm{SNF}(M)$. Compute a basis $\vec{w}_1, \ldots, \vec{w}_s$ of the left kernel of M.
For $i \leq s$, $\alpha'_i \leftarrow \prod_{j\leq s_0} \alpha_j^{w_{i,j}}$.
For $1 \leq i \leq r$, $\alpha'_{i+s} \leftarrow \alpha_{s_0+i}$ (the $(\alpha'_i)_{s<i\leq r+s}$ generate U_L).
return $\langle\alpha'_1\rangle \times \cdots \times \langle\alpha'_{s+r}\rangle, \mathbb{Z}/d'_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/d'_{s_0}$.

Proposition 5.3. *Algorithm 5 is correct and returns the S-class group and the S-unit group with probability $1 - 1/2^N$ where $N = [L;\mathbb{Q}]$ in time* $\mathrm{Poly}(\mathrm{Size}(S), \log(\Delta)) \cdot e^{\widetilde{O}(\sqrt{\log d})}$ *where $\Delta = \mathrm{disc}(L)$, and $d := \prod_{i\leq n} d_i$.*

References

[1] E. Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990), no. 191, 355–380. MR 1023756

[2] ———, *Improved approximations for Euler products*, Number theory (Halifax, 1994), CMS Conf. Proc., no. 15, Amer. Math. Soc., Providence, RI, 1995, pp. 13–28. MR 1353917

[3] Jens Bauch, Daniel J. Bernstein, Henry de Valence, Tanja Lange, and Christine van Vredendaal, *Short generators without quantum computers: the case of multiquadratics*, Advances in cryptology—EUROCRYPT 2017, I, Lecture Notes in Comput. Sci., no. 10210, Springer, 2017, pp. 27–59. MR 3652098

[4] J.-F. Biasse, M. Jacobson, and A. Silvester, *Algebraic techniques for number fields*, Second International Conference on Symbolic Computation and Cryptography, 2010, pp. 183–196.

[5] Jean-François Biasse, *Improvements in the computation of ideal class groups of imaginary quadratic number fields*, Adv. Math. Commun. **4** (2010), no. 2, 141–154. MR 2654130

[6] ———, *An $L(1/3)$ algorithm for ideal class group and regulator computation in certain number fields*, Math. Comp. **83** (2014), no. 288, 2005–2031. MR 3194139

[7] ———, *Subexponential time relations in the class group of large degree number fields*, Adv. Math. Commun. **8** (2014), no. 4, 407–425. MR 3290946

[8] Jean-François Biasse, Thomas Espitau, Pierre-Alain Fouque, Alexandre Gélin, and Paul Kirchner, *Computing generator in cyclotomic integer rings: a subfield algorithm for the principal ideal problem in $L_{|\Delta\mathbb{K}|}(\frac{1}{2})$ and application to the cryptanalysis of a FHE scheme*, Advances in cryptology—EUROCRYPT 2017, I, Lecture Notes in Comput. Sci., no. 10210, Springer, 2017, pp. 60–88. MR 3652099

[9] Jean-François Biasse and Claus Fieker, *Improved techniques for computing the ideal class group and a system of fundamental units in number fields*, ANTS X—Proceedings of the Tenth Algorithmic Number Theory Symposium, Open Book Ser., no. 1, Math. Sci. Publ., Berkeley, 2013, pp. 113–133. MR 3207410

[10] ———, *Subexponential class group and unit group computation in large degree number fields*, LMS J. Comput. Math. **17** (2014), no. A, 385–403. MR 3240816

[11] Jean-François Biasse and Michael J. Jacobson, Jr., *Practical improvements to class group and regulator computation of real quadratic fields*, Algorithmic number theory—ANTS IX, Lecture Notes in Comput. Sci., no. 6197, Springer, 2010, pp. 50–65. MR 2721412

[12] Jean-François Biasse, Michael J. Jacobson, Jr., and Alan K. Silvester, *Security estimates for quadratic field based cryptosystems*, Australasian Conference on Information Security and Privacy—ACISP 2010, Lecture Notes in Comput. Sci., no. 6168, Springer, 2010, pp. 233–247.

[13] Jean-François Biasse and Fang Song, *Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields*, Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2016, pp. 893–902. MR 3478440

[14] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system, I: The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265. MR 1484478

[15] Johannes Buchmann, *A subexponential algorithm for the determination of class groups and regulators of algebraic number fields*, Séminaire de Théorie des Nombres (Paris 1988/1989), Progr. Math., no. 91, Birkhäuser, Boston, 1990, pp. 27–41. MR 1104698

[16] J. P. Buhler, H. W. Lenstra, Jr., and Carl Pomerance, *Factoring integers with the number field sieve*, The development of the number field sieve, Lecture Notes in Math., no. 1554, Springer, 1993, pp. 50–94. MR 1321221

[17] D. Chatelain, *Bases des entiers des corps composés par des extensions quadratiques de* $\mathbb{Q}$, Ann. Sci. Univ. Besançon Math. (3) **1973** (1973), no. 6, 38. MR 0349625

[18] H. Cohen, F. Diaz y Diaz, and M. Olivier, *Subexponential algorithms for class group and unit computations*, J. Symbolic Comput. **24** (1997), no. 3-4, 433–441. MR 1484490

[19] Claus Fieker, Tommy Hofmann, and Carlo Sircana, *On the construction of class fields*, preprint, 2018.

[20] James L. Hafner and Kevin S. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc. **2** (1989), no. 4, 837–850. MR 1002631

[21] H. W. Lenstra, Jr., *On the calculation of regulators and class numbers of quadratic fields*, Journées arithmétiques, London Math. Soc. Lecture Note Ser., no. 56, Cambridge Univ. Press, 1982, pp. 123–150. MR 697260

[22] PARI Group, *PARI/GP version 2.9.4*, 2018.

[23] Michael Pohst (ed.), *Algorithmic methods in algebra and number theory*, Academic Press, London, 1987, Reprint of *J. Symbolic Comput.* **4**:1 (1987). MR 957703

[24] Sage Developers, *SageMath version 7.5.1*, 2017.

[25] Bernhard Schmal, *Diskriminanten,* $\mathbb{Z}$*-Ganzheitsbasen und relative Ganzheitsbasen bei multiquadratischen Zahlkörpern*, Arch. Math. (Basel) **52** (1989), no. 3, 245–257. MR 989879

[26] Daniel Shanks, *Class number, a theory of factorization, and genera*, 1969 Number Theory Institute, Proc. Sympos. Pure Math., no. 20, Amer. Math. Soc., Providence, RI, 1971, pp. 415–440. MR 0316385

[27] ———, *The infrastructure of a real quadratic field and its applications*, Proceedings of the Number Theory Conference (Boulder, CO, 1972), Univ. Colorado, Boulder, 1972, pp. 217–224. MR 0389842

[28] Denis Simon, *Équations dans les corps de nombres et discriminants minimaux*, Ph.D. thesis, Université Bourdeaux I, 1998.

[29] Arne Storjohann, *Algorithms for matrix canonical forms*, Ph.D. thesis, Swiss Federal Institute of Technology, 2000.

[30] Christoph Thiel, *On the complexity of some problems in algorithmic algebraic number theory*, Ph.D. thesis, Universität des Saarlandes, 1995.

[31] N. Tschebotareff, *Die Bestimmung der Dichtigkeit einer Menge von Primzahlen, welche zu einer gegebenen Substitutionsklasse gehören*, Math. Ann. **95** (1926), no. 1, 191–228. MR 1512273

Received 2 Mar 2018. Revised 21 Sep 2018.

JEAN-FRANÇOIS BIASSE: biasse@usf.edu
Department of Mathematics and Statistics, University of South Florida, Tampa, FL, United States

CHRISTINE VAN VREDENDAAL: c.v.vredendaal@tue.nl
NXP Semiconductors, Eindhoven, Netherlands

VOLUME EDITORS

Renate Scheidler
University of Calgary
Calgary, AB T2N 1N4
Canada

Jonathan Sorenson
Butler University
Indianapolis, IN 46208
United States

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

ISBN: 978-1-935107-02-6 (print), 978-1-935107-03-3 (electronic)

First published 2019.

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 2

Thirteenth 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 thirteenth ANTS meeting, held July 16-20, 2018, at the University of Wisconsin-Madison. It includes revised and edited versions of 28 refereed papers presented at the conference.

Edited by Renate Scheidler and Jonathan Sorenson

CONTRIBUTORS

Simon Abelard
Sonny Arora
Vishal Arul
Angelica Babei
Jens-Dietrich Bauch
Alex J. Best
Jean-François Biasse
Alin Bostan
Reinier Bröker
Nils Bruin
Xavier Caruso
Stephanie Chan
Qi Cheng
Gilles Christol
Owen Colman
Edgar Costa
Philippe Dumas
Kirsten Eisenträger
Claus Fieker
Shuhong Gao
Pierrick Gaudry
Alexandre Gélin
Alexandru Ghitza
Laurent Grémy
Jeroen Hanselman
David Harvey
Tommy Hofmann
Everett W. Howe
David Hubbard
Kiran S. Kedlaya
Thorsten Kleinjung
David Kohel
Wanlin Li
Richard Magner
Anna Medvedovsky
Michael Musty
Ha Thanh Nguyen Tran
Christophe Ritzenthaler
David Roe
J. Maurice Rojas
Nathan C. Ryan
Renate Scheidler
Sam Schiavone
Andrew Shallue
Jeroen Sijsling
Carlo Sircana
Jonathan Sorenson
Pierre-Jean Spaenlehauer
Andrew V. Sutherland
Nicholas Triantafillou
Joris van der Hoeven
Christine Van Vredendaal
John Voight
Daqing Wan
Lawrence C. Washington
Jonathan Webster
Benjamin Wesolowski
Yinan Zhang
Alexandre Zotine