Quantum Algorithms for many-to-one Functions to Solve the Regulator and the Principal Ideal Problem

Arthur Schmidt

Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

Abstract. We propose new quantum algorithms to solve the regulator and the principal ideal problem in a real-quadratic number field. We improve the algorithms proposed by Hallgren ([Hal02b], [Hal07]) by using two different techniques. The first improvement is the usage of a period function which is not one-to-one on its period. We show that even in this case Shor's algorithm computes the period with constant probability. The second improvement is the usage of reduced forms (a, b, c) of discriminant Δ with $a > 0$ instead of reduced ideals of the same discriminant. These improvements reduce the number of required qubits by at least $2 \log \Delta$.

1 Introduction

Quantum algorithms can be used to achieve a sub-exponential or even exponential speed-up over known classical algorithms for some mathematical problems by using Shor's quantum framework. Shor's algorithms for factoring and solving the discrete logarithm problem [Sho94] have been adapted to different problems. The computation of the regulator (Regulator Problem) of a real-quadratic number field and the solution of the principal ideal problem (PIP) are two examples of such adaptions. Classically, these problems can be solved in sub-exponential time assuming the generalized Riemann hypothesis (GRH). For the quantum world, polynomial time algorithms were proposed by Hallgren in [Hal02b].

Regulator computation and the PIP are interesting problems not only from a pure mathematical point of view. In [BW90], Buchmann and Williams proposed a Diffie-Hellman-like cryptosystem which security is based on PIP. Thus, if we could solve the PIP, we can break the cryptosystem from [BW90].

The regulator computation differs from all the other settings where Shor's algorithm can be applied. It operates on a structure (the infrastructure of principal reduced ideals) which is not a group, since it lacks the associativity. However, Hallgren showed that Shor's algorithm can still be used in this case.

RP and PIP require the computation of natural logarithms. Thus, one problem which arises during these computations is the choice of the right approximation of natural logarithms. There is no known way to choose the approximation a priori for a given number field. Thus, the functions proposed in [Hal02b] cannot be computed in polynomial time. This problem was solved by Schmidt and Vollmer in [SV05] by using non-canonical number theoretic constructions, and by Hallgren himself in [Hal07], by defining functions which are periodic only on a subset of the possible function values. In our paper we show that this problem can also be solved by using functions which are always periodic but are many-to-one on their fundamental period. We show that Shor's framework computes the right period even in such a case with constant success probability. We obtain a Monte Carlo type algorithm which does not depend on GRH.

The problem to compute the period of a function which is not one-to-one was first addressed by Boneh and Lipton in [BL95]. The authors presented algorithms for functions in $\mathbb{Z}$ which have integer periods. In [ME99], Mosca and Eckert generalized this result for finitely generated Abelian groups for some restricted class of functions. In [HH00] and [Hal02a], these restrictions were eliminated by Hales and Hallgren. In our paper, we solve this problem for certain many-to-one functions whose periods are irrational.

There are two equivalent languages which can be used to describe elements of and problems in quadratic number fields. The first is the language of ideals, which is usually used for formal definitions of the underlying concepts and elements of a number field. The second is the language of quadratic forms, which is used to describe algorithms and carry out computations. In this paper, we will use both languages in exactly such a way.

Our contribution in this paper is the following. We present more efficient versions of algorithms for computing the regulator and solving the PIP. Since the PIP problem is a basis for a cryptosystem, it gives

us a better tool to compare this cryptosystem to others in their resistance against quantum attacks. Thus, we can make a better choice which cryptosystem should be used if we assume that quantum computers of a certain size can be build ([Sch06], [Sch07]). The second contribution are examples for problems which solution can be improved by using functions which are not one-to-one on their fundamental periods. In this paper, we do not do a full analysis of the number of qubits for the presented algorithms. Instead, we only reduce the complexity of certain parts of the known algorithms. A first complete analysis of the algorithms was presented in [Sch07]. We will improve this analysis by using more efficient algorithms in a subsequent paper.

Our paper is organized as follows. In the next section, we give a short overview of the quantum framework. In section 3, we present the necessary background from number theory. In section 4, we describe a quantum algorithm for computing the regulator of a given number field. In section 5, we present an algorithm for solving the principal ideal problem. We summarize our results and describe open problems in the last section.

2 Quantum Computing Background

Many polynomial time quantum algorithms that solve problems for which only sub-exponential or even exponential classical algorithms are known use the (inverse) quantum Fourier transform (QFT) as a subroutine. The problems in this class can be reduced to the problem of finding a basis for a period lattice Λ of an appropriate function[1]. For example, Shor's factoring algorithms computes the factors of an integer n by determining the period of the function $f(x) = a^x \mod n$, with $1 < a < n$. The period of f is the order of a in the finite abelian group $(\mathbb{Z}/n\mathbb{Z})^*$, and the corresponding lattice is $\text{order}(a)\mathbb{Z}$. The objective of the quantum subroutine is to find an approximation of a basis B for the dual lattice Λ^*. During the classical post-computation step the basis B is used to compute a basis of the original lattice Λ. The latter task can be done by using a continued fraction expansion as proposed in Shor's original paper [Sho94], by using a simultaneous Diophantine approximation as proposed by Seifert in [Sei01], or by using techniques by Buchmann and Pohst ([BP89], [BK93]) as proposed by Hallgren in [Hal05] and Schmidt and Vollmer in [SV05].

The framework for such an algorithm is the following. The quantum computer uses two registers: one to store the input vector of the function and one to store the function value. The algorithm starts by creating a superposition of all possible states in the first register, by computing the function value to the second register, and by measuring the second register. By the laws of quantum mechanics, the measurement changes the state of the quantum computer to $\sum_{v \in L} |k + v\rangle |f(k)\rangle$ where k is a random vector and L is a subset of Λ. Next, the QFT and a measurement is applied to the first register. One useful property of the set computed by the QFT is that it is independent of the coset $k + \Lambda$. Thus, QFT always creates a superposition of values which approximate the basis of Λ^* independent of k. The other useful property of the QFT is that the elements in the superposition are almost uniformly distributed. These two properties imply that, for a fixed dimension of the lattice, an approximation of the basis B is computed with a constant probability after running the above algorithm a constant number of times.

In the following sections, we will define periodic functions whose period lattices can be used to compute the regulator and to solve the PIP resp. DL-problem.

3 Number Theory Background

3.1 Ideals

Let Δ be a positive integer which is not a square such that $\Delta \equiv 0, 1 \mod 4$. Then the module $\mathcal{O}_\Delta = \mathbb{Z} + \frac{\Delta + \sqrt{\Delta}}{2}\mathbb{Z}$ is a real-quadratic order. The field of fractions of the order $\mathcal{O}_\Delta$ is the real-quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$. An element $\alpha \in \mathbb{Q}(\sqrt{\Delta})$ can be written as $\alpha = a + b\sqrt{\Delta}$ with $a, b \in \mathbb{Q}$. The norm of α is $N(\alpha) = a^2 - b^2\Delta$.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two subsets of $\mathcal{K}$, then the product $\mathcal{X}\mathcal{Y}$ is the additive subgroup of $\mathcal{K}$ generated by $\{xy \mid x \in \mathcal{X}, \ y \in \mathcal{Y}\}$. An integral $\mathcal{O}_\Delta$-ideal is a module $\mathfrak{a} \subseteq \mathcal{O}_\Delta$ such that $\mathfrak{a}\mathcal{O}_\Delta \subseteq \mathfrak{a}$. A (fractional) ideal $\mathfrak{a}$ is a subset of $\mathcal{K}$ such that $d\mathfrak{a}$ is a integral ideal for a $d \in \mathbb{Z}$. An ideal $\mathfrak{a}$ is invertible, if there exists an ideal $\mathfrak{b}$ with $\mathfrak{a}\mathfrak{b} = \mathcal{O}_\Delta$. By $\mathcal{I}$, we denote the set of invertible ideals.

---

[1] We say that a function $f : \mathbb{R}^n \to S$ has a period lattice $\Lambda \subset \mathbb{R}^n$ if Λ is a lattice and $f(x) = f(x + \lambda)$ for all $\lambda \in \Lambda$

Each ideal $\mathfrak{a}$ has the form

$$\mathfrak{a} = q(a\mathbb{Z} + \frac{b + \sqrt{\Delta}}{2}\mathbb{Z}),$$

where $a, b \in \mathbb{Z}$, $q \in \mathbb{Q}$, $a, q > 0$, b is unique modulo $2a$, $c = (b^2 - \Delta)/(4a) \in \mathbb{Z}$, and $\gcd(a, b, c) = 1$. The ideal is called reduced, if $a > 0$ and $|\sqrt{\Delta} - 2|a|| < b < \sqrt{\Delta}$. By $\mathcal{R}$ we denote the set of reduced ideals.

Two ideals $\mathfrak{a}$ and $\mathfrak{b}$ are equivalent if there is $\alpha \in \mathcal{K}$ such that $\mathfrak{b} = \alpha\mathfrak{a}$. The set of equivalence classes of ideals forms a finite abelian group under ideal multiplication. We denote this group by Cl_Δ We have $\mathrm{Cl}_\Delta = \mathcal{I}/\mathcal{P}$, where $\mathcal{P} = \{\alpha\mathcal{O}_\Delta \mid \text{with } \alpha \in \mathcal{K}\}$ is the set of principal ideals.

Every ideal $\mathfrak{a}$ is equivalent to a reduced ideal. The equivalent reduced ideal can be computed by applying the reduction operator $\rho(\mathfrak{a}) = \gamma\mathfrak{a}$, with $\gamma = -2c/(q(b + \sqrt{\Delta}))$, at most $\log_2(a/\sqrt{\Delta}) + 2$ times.

By theorem of Dirichlet, every unit of $\mathcal{O}_\Delta$ can be written as $\pm\epsilon^k$ with an integer k and a fundamental unit ϵ. It is easy to see that the norm of every unit is equal to plus or minus one.[2] In general, the number of bits which are necessary to represent a unit is exponential (in $\log \Delta$). Thus, instead of computing a fundamental unit ϵ we compute the regulator defined as $R = \ln|\epsilon|$. If we confine ourself to units with norm plus one, then there is a fundamental unit ϵ' of norm one such that every unit of norm one has the form $\pm(\epsilon')^k$. In this case, $R^+ = \ln|\epsilon'|$ is called the regulator in the narrow sense. Note that in a number field either $R = R^+$ or $R = R^+/2$. In our computations we will only consider the narrow case.

Principal ideals can be ordered on a circle of circumference R by using the distance function $\delta : \mathcal{P} \to \mathbb{R}/R\mathbb{Z}$: $\alpha\mathcal{O}_\Delta \mapsto \mathrm{Log}\,\alpha$ with $\mathrm{Log}\,\alpha = \frac{1}{2}\ln|\sigma(\alpha)/\alpha| \mod R$. Note that the unit ideal has distance zero. The distance between two ideals $\mathfrak{a}$ and $\mathfrak{b}$ is defined by $\delta(\mathfrak{a}, \mathfrak{b}) = \delta(\mathfrak{a}) - \delta(\mathfrak{b}) \mod R^+$. It has two important properties: $1/\sqrt{\Delta} < \delta(\mathfrak{a}, \rho(\mathfrak{a})) < \ln\sqrt{\Delta}$ and $\delta(\mathfrak{a}, \rho(\rho(\mathfrak{a}))) > \ln 2$ for all reduced ideals $\mathfrak{a}$. There is a minimal positive integer k such that the sequence $(\mathcal{O}_\Delta, \rho(\mathcal{O}_\Delta), \ldots, \rho^k(\mathcal{O}_\Delta) = \mathcal{O}_\Delta)$ contains all principal reduced ideals. Thus, by applying ρ we can "walk" through all these ideals. The product of all γ's which occur during the computation of ρ is a fundamental unit.

For an $x \in \mathbb{R}$ and a principal ideal $\mathfrak{a} = \alpha\mathcal{O}_\Delta$, we define $\delta(\mathfrak{a}, x) = x - \mathrm{Log}\,\alpha \mod R$. Let $\mathfrak{a} \in \mathcal{P}$ be such that $\delta(\mathfrak{a}, x) \leq 0$ and $\delta(\rho(\mathfrak{a}), x) > 0$, then we say that the ideal is left of or at x and denote it by $\mathfrak{a}_-(x)$. The computation of $\mathfrak{a}_-(x)$ requires the computation of natural logarithms. We cannot do this exactly. Moreover, to the best of our knowledge, the computation $\mathrm{Log}\,\alpha$ to any a priori fixed precision does not allow to correctly make the decision for some x's whether $\delta(\mathfrak{a}, x) \leq 0$ or $\delta(\mathfrak{a}, x) > 0$. If, however, we successively increase the precision to break a tie, we might spend an amount of time on this single computation that exceeds any a priori given polynomial bound for the run-time of the total algorithm.[3] Therefore, in our algorithms, we only approximate natural logarithms. For an $x \in \mathbb{Q}$, this approach produces some $\tilde{\mathfrak{a}}_-(x)$ which is left of or at x according to these approximative logarithm computations. We take into account that for some x's $\mathfrak{a}_-(x) \neq \tilde{\mathfrak{a}}_-(x)$.

In the rest of the section, we consider quadratic forms, show their correspondence to ideals, and describe the advantage to use them in our algorithms.

3.2 Quadratic Forms

An integer indefinite quadratic form of discriminant Δ is a polynomial $aX^2 + bXY + cY^2$, where $a, b, c \in \mathbb{Z}$, $\gcd(a, b, c) = 1$, and $\Delta = b^2 - 4ac > 0$. If Δ is not a square, then the form is irreducible. The form is reduced if $|\sqrt{\Delta} - 2|a|| < b < \sqrt{\Delta}$. It is easy to see that if (a, b, c) is reduced, then $ac < 0$.

There is a well known bijection (see [BV07], Theorem 4.4.4) between invertible ideals and Γ-Orbits[4] of irreducible indefinite forms with positive a. This bijection maps distances of ideals to distances of forms. Similarly to the ideal case, we can "walk" on the principal circle by applying the ρ-operator to the form $f = (a, b, c)$ which is $\rho(f) = (c, B, A)$ such that $B \equiv -b \mod 2c$, $|\sqrt{\Delta} - 2|c|| < B < \sqrt{\Delta}$ and $A = (B^2 - \Delta)/(4c)$. The difference to the ideal case is that here, the sign of the first coefficient alternates whereas in the ideal case it is always positive. In our computations, we use this fact and look at reduced principal forms (a, b, c) left of

---

[2] Note that there is well know connection between fundamental units and solutions of the famous Pell equation (see [JW09] for more information about it).

[3] This is exactly the point where there remains a gap in Hallgrens proof of polynomial run-time of his algorithm for the quadratic case.

[4] A Γ-Orbit of a form (a, b, c) is the set $\{(a, B, C) \mid b \equiv B \mod 2a \text{ and } C = (B^2 - \Delta)/4a\}$.

or at x with the additional condition that $a > 0$. We denote the set of reduced principal forms with positive a by $\mathcal{R}^+$. The advantage in using forms from $\mathcal{R}^+$ over all reduced forms is the following. As mentioned above, the distance between an ideal $\mathfrak{a}$ and $\rho^2(\mathfrak{a})$ is at least $\ln 2$. This implies that the distance between two forms from $\mathcal{R}^+$ is at least $\ln 2$, too. In contrast, the distance between forms in the set of all principal reduced forms is at least $1/\sqrt{\Delta}$. Thus, by using $\mathcal{R}^+$, we have the property that the minimum distance between two forms is independent of Δ.

In our algorithms, we have to compute forms left of or at x with $x > \Delta$. Since $\delta(\mathfrak{a}, \rho(\mathfrak{a})) < \log \sqrt{\Delta}$, the time complexity of this computation is exponential in $\log \Delta$. To "jump" over larger distances, we use giant steps which consist of form composition and reduction. Let $f = (a, b, c)$ be the composition of two forms (resp. ideals) $f_1 = (a_1, b_1, c_1)$ and $f_2 = (a_2, b_2, c_2)$. Form f has coefficients $a = a_1 a_2 / m$, $b = (j a_2 b_1 + k a_1 b_2 + l(b_1 b_2 + \Delta)/2)/m \mod 2a$, where $j a_2 + k a_1 + l(b_1 + b_2)/2 = m = \gcd(a_1, a_2, (b_1 + b_2)/2)$, and $c = (b^2 - \Delta)/(4a)$. Form f is in general not reduced, so by applying ρ at most $\log \sqrt{\Delta} + 2$ times we obtain a reduced form which is equivalent to the composition of f_1 and f_2. Let k be the number of ρ-applications. For the distances, we have the following equation:

$$\delta(f_1 * f_2) = \delta(\rho^k(f)) = \delta(f_1) + \delta(f_2) + \delta', \tag{1}$$

where $\delta' = \delta(f, \rho^k(f))$ is small (at most $\pm \ln \Delta$). An ideal composition followed by a reduction imply a structure which is almost a group (since, in general, $\delta' \neq 0$ it is not exactly a group), we call it the infrastructure (see [Len82], [BV07], or [JW09] for more details).

In our algorithm we compute the form $\tilde{g}_{x/4}$ left of or at $x \in (1/4)\mathbb{Z}$ using an approximate logarithm computation. This can be done as follows. Let g be the unit form. We first compute the form $h = \rho(\rho(g))$. We know that $\delta(g, h) > \ln 2$. Thus, we can use a square-and-multiply method to compute the form $\tilde{g}_{x/4}$. We need to estimate the number of operations (squares, multiplications, reduction) to determine a necessary logarithm precision. Since in our algorithms $x < \Delta^2$, the number of squares and multiplication is at most $2(2 \log_2 \Delta + 2)$. Each square and multiplication is followed by $\log \sqrt{\Delta} + 2$ reductions. Therefore, the total number of operations is at most $(c \log_2 \Delta)$, where $c < 10$ is a constant. If we choose the precision of each logarithm computation to be at least $1/(8c \log \Delta)$, then, by (1), we obtain $|\delta(\tilde{g}_{x/4}) - \tilde{\delta}(\tilde{g}_{x/4})| < 1/8$, where $\tilde{\delta}$ is the approximation of δ computed by the above algorithm. This approximation is required in the subsequent sections. The computation of $\tilde{g}_{x/4}$ can be done in time polynomial in $\log \Delta$, since all the computations (square, multilication, reduction, and logarithm evaluations with the necessary precision) can be done in polynomial time.

4 Computing the Regulator

In this section we solve the regulator problem which is defined as follows.

Definition 1 (Regulator Problem). *Given Δ, find an integer R' with $|R' - R^+| < 1$ where R^+ is the regulator of $\mathbb{Q}(\sqrt{\Delta})$.*

We first give the definition of the periodic function for computing the regulator.

Definition 2. *Fix an algorithm $\tilde{\ln}$ for computing an approximation of the natural algorithm. The function*

$$\mathrm{Reg} : \mathbb{Z} \to \mathcal{R}^+ : x \longmapsto \tilde{g}_{x/4}$$

maps an integer x to the principal reduced form $\tilde{g}_{x/4} = (a, b, c)$, $a > 0$, such that, with respect to $\tilde{\ln}$, $\tilde{g}_{x/4}$ is left of or at $x/4$. The precision of $\tilde{\ln}$ must be chosen such that, for all x, $|\delta(\tilde{g}_{x/4}) - \tilde{\delta}(\tilde{g}_{x/4})| < 1/8$, where $\tilde{\delta}$ is the approximation of δ which uses $\tilde{\ln}$ instead of $\ln$.

In the next two lemmas, we will show that Reg is periodic. In Lemma 2, we will show that for every $g \in \mathcal{R}^+$ there are areas of successive integers in every period of Reg which are all mapped to g, that the number of integers in these areas is at most $\ln \Delta + 3$, and that this number differs by at most 4 in different periods. In Lemma 1, we will show that the areas are non-empty and the first element occurs with a period $\approx 4R^+$

Lemma 1. *For every $g \in \mathcal{R}^+$, there is a $y = 4\delta(g) + 1/2$ such that*

$$\forall k \in \mathbb{Z}. \exists \epsilon \in \mathbb{R}, |\epsilon| \leq 1.(x = y + 4kR^+ + \epsilon \in \mathbb{Z}, \ \mathrm{Reg}(x) = g, \ \text{and} \ \mathrm{Reg}(x-1) = \rho^{-2}(g)).$$

Proof. Let $g \in \mathcal{R}^+$, $y = 4\delta(g) + 1/2$, $k \in \mathbb{Z}$, and $x \in \mathbb{Z}$, such that $x/4 = \delta(g) + kR^+ + \delta$ with $-1/8 \leq \delta < 1/8$

From $\ln 2 < \delta(\rho^2(g)) - \delta(g)$, we obtain $\tilde{\delta}(\rho^2(g)) - \tilde{\delta}(g) > \ln 2 - 1/4 > 1/4$. That means that for every g there is at least one x in each period with $\mathrm{Reg}(x) = g$ and the period lattice of Reg has no gaps.

Now assume $-1/8 \leq \delta \leq 0$. In this case we have $x/4 \leq \delta(g) + kR^+ \leq x/4 + 1/8$ and therefore $x/4 - 1/8 < \tilde{\delta}(g) + kR^+ < (x+1)/4$. This implies that $\mathrm{Reg}(x-1) = \rho^{-2}(g)$, $\mathrm{Reg}(x+1) = g$, and $\mathrm{Reg}(x) \in \{\rho^{-2}(g), g\}$. If $\mathrm{Reg}(x) = \rho^{-2}(g)$, then $|x - y - 4kR^+| \leq 1/2$. If $\mathrm{Reg}(x) = \rho^{-2}(g)$, then $|(x+1) - y - 4kR^+| \leq 1$. Thus in both cases the ϵ, as defined in the lemma, exists.

The case $0 < \delta < 1/8$ is analogous. $\square$

Lemma 2. *Let Δ be a discriminant of a real-quadratic number whose regulator R^+ is greater than $5 \ln \Delta$. Let $g \in \mathcal{R}^+$, $y = 4\delta(g) + 1/2$, $k \in \mathbb{Z}$, and $\epsilon_{(g,k)} \in \mathbb{R}$, be defined as in the last lemma. Then there exists an $m_{(g,k)} \in \mathbb{Z}$, $1 \leq m_{(g,k)} < \ln \Delta + 3$, such that the following is true:*

1. $\mathrm{Reg}(y + 4kR^+ + \epsilon_{(g,k)} + m_{(y,k)} + 1) = \rho^2(g)$.
2. $\mathrm{Reg}(y + 4kR^+ + \epsilon_{(g,k)} + m) = g$, *for all $m \in \mathbb{Z}$, $0 \leq m \leq m_{(g,k)}$*
3. $\max_{k,k' \in \mathbb{Z}} |m_{(g,k)} - m_{(g,k')}| \leq 4$.

Proof. We first prove the existence of $m_{(y,k)} \in \mathbb{Z}$, $0 \leq m_{(y,k)} < \ln \Delta + 3$ such that (1) is satisfied. This follows from

$\delta(\rho^2(g)) - \delta(g) \leq \ln \Delta$ and the assumption that $R^+ > 5 \ln \Delta$, which implies $\mathrm{Reg}(y + 4kR^+ + \epsilon_{(y,k)}) \neq \mathrm{Reg}(y + 4kR^+ + \epsilon_{(y,k)} + \lceil \ln \Delta \rceil + 2)$.

(2) and (3) follow easily from the fact that we look for ideals left of or at a multiple of $1/4$ and the approximation quality of function Reg is at least $1/8$. $\square$

Now we present our algorithms. We first start with the quantum subroutine.

Algorithm 1 REGULATOR-DUAL

Input: Discriminant Δ, q which is a power of two and $q/2 \leq 5\Delta(\ln \Delta)^2 < q$.
Output: Approximation of a number from $(q/R^+)\mathbb{Z}$.

1. (initial state) $|0\rangle, |(1, \Delta \mod 2)\rangle$.
2. (create superposition) $\longrightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle, |(1, \Delta \mod 2)\rangle$.
3. (compute Reg) $\longrightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle, |\mathrm{Reg}(x)\rangle$.
4. (measure the second register)

$$\longrightarrow \frac{1}{\sqrt{p}} \sum_{k \in \mathcal{M}} \sum_{m=0}^{m_{(x',k)}} |x' + 4R^+k + m + \epsilon_{(x',k)}\rangle, |\mathrm{Reg}(x')\rangle$$

with a random $x' \in \{0, \ldots, \lfloor 4R^+ \rfloor\}$, $\epsilon_{(x',k)}$ and $m_{(x',k)}$ as defined in lemma 2, $\mathcal{M} = \mathcal{M}_{x'} = \{k \in \mathbb{Z} \mid 0 \leq x' + 4R^+k + \epsilon_{(x',k)} < q\}$ and $p = \mathrm{card} \{x \in \mathbb{Z} \mid 0 \leq x < q$ and $\mathrm{Reg}(x) = \mathrm{Reg}(x')\}$.
5. (apply quantum Fourier transform to the first register)

$$\longrightarrow \frac{1}{2\sqrt{pq}} \sum_{y=0}^{4q-1} \sum_{k \in \mathcal{M}} \sum_{m=0}^{m_{(x',k)}} \exp\left(2\pi i \frac{x' + 4R^+k + m + \epsilon_{(x',k)}}{4q} y\right) |y\rangle, |\mathrm{Reg}(x')\rangle.$$

6. Measure and return the first register y.

Theorem 1. *Let Δ be a discriminant of a real-quadratic number field whose regulator is at least $32 \ln \Delta$. The algorithm* REGULATOR-DUAL *computes an approximation of a random element from $(q/R^+)\mathbb{Z}$. The*

approximation has the form $(q/R^+)z + \omega$ where $z \in \mathbb{Z}$ and $|\omega| \leq 1/2$. The algorithm succeeds with probability at least 2^{-11} and requires at most $2\log(\Delta) + 2\log\ln\Delta + N + 7$ qubits, where N is the number of temporary qubits which are necessary to execute operations on forms to compute Reg.[5]

Proof. We use the same notation as in the theorem and algorithm. Let $m_{max} = \max_{k \in \mathcal{M}_{x'}} m_{(x',k)}$, $m_{min} = \min_{k \in \mathcal{M}_{x'}} m_{(x',k)}$, and

$$\mathcal{Y} = \{\, y \in \mathbb{Z} \mid 0 \leq y \leq \frac{q}{4(m_{max}+1)} \text{ and } \frac{y}{4q} = \frac{z}{4R^+} + \omega_y \text{ with } z \in \mathbb{Z} \text{ and } |\omega_y| \leq \frac{1}{8q} \,\}. \tag{2}$$

The probability to measure a $y \in \mathcal{Y}$ is

$$\Pr(y \in \mathcal{Y}) = \frac{1}{4pq} \left| \sum_{k \in \mathcal{M}} \sum_{m=0}^{m_{(x',k)}} \exp\left(2\pi i \frac{4R^+k + m + \epsilon_{(x',k)}}{4q} y \right) \right|^2 .$$

Since we have

$$\frac{(4R^+k + m + \epsilon_{(x',k)})y}{4q} = 4kR^+(\frac{z}{4R^+} + \omega_y) + \frac{m + \epsilon_{(x',k)}}{4q}y \equiv 4R^+k\omega_y + \frac{(m + \epsilon_{(x',k)})y}{4q}$$

modulo 1 and since the function exp is periodic, we can write

$$\Pr(y \in \mathcal{Y}) = \frac{1}{4pq} \left| \sum_{k \in \mathcal{M}} \sum_{m=0}^{m_{(x',k)}} \exp\left(2\pi i (4R^+k\omega_y + \frac{(m + \epsilon_{(x',k)})y}{4q}) \right) \right|^2 . \tag{3}$$

By Lemma 1, 2, and Equation (2), we follow $|4R^+k\omega_y| \leq 1/8$ and $-1/16 \leq (m + \epsilon_{(x',k)})y/(4q) \leq 1/16$. This means that (3) is a sum of p vectors of length one which all lie in a segment of size $\pi/2$. Thus, the probability that we measure a certain $y \in \mathcal{Y}$ is

$$\Pr(y \in \mathcal{Y}) \geq \frac{1}{4pq} \left| p\frac{\sqrt{2}}{2} \right|^2 = \frac{p}{8q}.$$

Next we approximate the lower bound for p and the cardinality of $\mathcal{Y}$. We have

$$p \geq (\mathrm{card}\,\mathcal{M}_{x'} - 1)(m_{min} + 1) + 1 \geq \left(\frac{q}{4R} - \frac{9}{4} \right)(m_{min} + 1) + 1 \geq \frac{q}{8R}(m_{min} + 1)$$

$$\mathrm{card}\,\mathcal{Y} \geq \{\, z \in \mathbb{Z} \mid 1 \leq z \leq \frac{R}{4(m_{max}+1)} - \frac{R}{2q} \,\} \geq \frac{R}{4(m_{max}+1)} - \frac{3}{2} \geq \frac{R}{8(m_{max}+1)}.$$

The condition $R^+ > 32\ln\Delta$ ensures that the set $\mathcal{Y}$ contains at least three different elements. Thus, we have

$$\sum_{y \in \mathcal{Y}} \Pr(y \in \mathcal{Y}) \geq \frac{p}{8q} \mathrm{card}\,\mathcal{Y} \geq \frac{m_{min} + 1}{2^9(m_{max}+1)} \geq \frac{1}{2^{11}}.$$

The number of qubits can be determined as follows. The first register requires at most $\log\Delta + 2\log(\ln\Delta) + 5$ qubits to keep $q < 10\Delta(\ln\Delta)^2$. For the second register, $\log\Delta + 2$ qubits are necessary to keep the coefficients a and b of the form (a, b, c). Since Δ is fixed, it is not necessary to store c. Since $(a, b, c) \in \mathcal{R}^+$ is reduced, we have $0 < a, b \leq \sqrt{\Delta}$. □

On the next page, we present the complete algorithm for computing the regulator based on the quantum subroutine described above. We have the following theorem.

Lemma 3. *Let $q > (R^+)^2$ and y_i, z_i be defined as in* REGULATOR, *then we have $|y_1/y_2 - z_1/z_2| \leq 1/(2z_2^2)$.*

---

[5] In [Sch07], it it shown that $N < 10.5\log\Delta + O(\log^2(\log\Delta))$

Algorithm 2 REGULATOR

Input: A discriminant Δ of a real-quadratic field $\mathcal{K}$.
Output: The regulator R^+ of $\mathcal{K}$.

1. Test classically whether $R^+ < 32 \ln \Delta$. If the answer is yes, compute classically the required approximation of R^+ and go to 4.
2. Use REGULATOR-DUAL to compute $y_1 = (q/R^+)z_1 + \omega_1$ and $y_2 = (q/R^+)z_2 + \omega_2$, $|\omega_1|, |\omega_2| \leq 1/2$, which approximate random vectors in $(q/R^+)\mathbb{Z}$.
3. W.l.o.g. assume $y_1 \leq y_2$. Use the continued fraction expansion algorithm applied to y_1/y_2 to compute z_1 and z_2. The number qz_1/y_1 is an approximation of the regulator which can be improved classically.
4. Return the approximation R^+.

Proof. We have the following inequality

$$\left| \frac{y_1}{y_2} - \frac{z_1}{z_2} \right| \leq \left| \frac{qz_1 + R^+\omega_1}{qz_2 + R^+\omega_2} - \frac{z_1}{z_2} \right| \leq \frac{R^+}{2} \left| \frac{z_1 + z_2}{z_2(qz_2 + R^+\omega_2)} \right| \leq \frac{R^+}{qz_2 - R^+/2} \leq \frac{1}{2z_2^2}.$$

The last inequality is true because of the choice of $q > (R^+)^2$ and $y \in \mathcal{Y}$ with $\mathcal{Y}$ from (2). $\qquad\square$

Theorem 2. REGULATOR *computes an approximation of the regulator R^+ of a real-quadratic number field $\mathbb{Q}(\sqrt{\Delta})$ in quantum-polynomial time $O(\mathrm{polylog}(\log \Delta))$. It is a Monte Carlo type algorithm which succeeds with probability at least 2^{-26}. The algorithm requires at most $2\log(\Delta) + 2\log \ln \Delta + N + 7$ qubits, where N is the number of temporary qubits which are necessary to execute operations on forms to compute* Reg.

Proof. We use the same notation as in the theorem and the algorithm.

First, assume $R^+ < 32 \ln \Delta$. In this case, the regulator can be computed completely classically by using the polynomial time algorithm from [BB94].

Next, assume $R^+ > 32 \ln \Delta$. In this case the cardinality of $\mathcal{Y}$ from (2) is at least 3. Thus, by running REGULATOR-DUAL twice we obtain two different non-zero $y_1, y_2 \in \mathcal{Y}$ with probability at least $(1/8)2^{-11}2^{-11} = 2^{-25}$. Since $|\mathrm{Cl}_\Delta|R^+ < \sqrt{\Delta}(\ln \sqrt{\Delta} + 1)/2$ (see [Hua82]), we have $(4R^+)^2 < \Delta(\ln \sqrt{\Delta} + 1)^2 < q$. Therefore Lemma 3 holds and we can apply the continued fraction expansion algorithm to y_1 and y_2 to compute z_1 and z_2 (assuming $\gcd(z_1, z_2) = 1$ which is true with probability at least $6/\pi^2$). The number qz_1/y_1 is an approximation of the regulator which can be improved classically ([BB94] [Mau00]). The success probability of the algorithm is at least $(6/\pi^2)2^{-25} > 2^{-26}$

The number of qubits follows directly from Theorem 1. $\qquad\square$

5 Solving the Principal Ideal Problem

In this section, we present an algorithm for solving the principal ideal problem and the discrete logarithm problem in the infrastructure of a real-quadratic number field.

Definition 3 (Principal ideal problem). *Given a reduced form g, decide whether g is principle and, if so, find $\delta(g)$.*

To solve the PIP, we extend the function Reg to the following one.

Definition 4. *Let g be a reduced principal form. Fix an algorithm $\tilde{\ln}$ for computing an approximation of the natural algorithm. All the distance operations δ below are carried out with this $\tilde{\ln}$. The function*

$$\mathrm{PIP} : \mathbb{Z} \times \mathbb{Z} \to \mathcal{R}^+ \; : \; (x, y) \longmapsto \tilde{g}_{(x,y)}$$

maps two integers x and y to a reduced principal form $\tilde{g}_{(x,y)} = (a, b, c)$, $a > 0$, left of or at $\delta(g^x) + y/4$. The precision of $\tilde{\ln}$ must be chosen such that $|\delta(\tilde{g}_{(x,y)}) - \tilde{\delta}(\tilde{g}_{(x,y)})| < 1/8$. for all x and y.

The next lemma is an extension of Lemmas 1 and 2.

Lemma 4. *Let n be the smallest positive integer such that $g^n \sim \mathcal{O}_\Delta$. Let $S = \text{dist}(\mathcal{O}_\Delta, \mathfrak{b}^n)$ and Λ be the lattice generated by $((n, -S)^t, (0, R^+)^t)$.[6] Then for all $(x_1, x_2), (x_1', x_2') \in \mathbb{Z}^2$, there exist an $\epsilon_{(\mathbf{x}, x_2')}$, $|\epsilon_{(\mathbf{x}, x_2')}| < 1$, and $1 \leq m_{(\mathbf{x}, x_2')} \leq \ln \Delta + 3$ such that $-4x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x}, x_2')} \in \mathbb{Z}$ and*

$$\text{PIP}(x_1' + x_1 n, x_2' - x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x}, x_2')}) = \text{PIP}(x_1', x_2')$$

iff $(x_1, x_2) \in \Lambda$ and $0 \leq m \leq m_{(\mathbf{x}, x_2')}$. As in Lemma 2, we have $max_{\mathbf{x} \in \Lambda} |m_{(\mathbf{x}, x_2')}| \leq 4$. □

Lattice Λ is the period lattice of PIP. Let Λ^* be the lattice dual to Λ. It is easy to see that

$$\begin{pmatrix} 1/n & 0 \\ s/(4nR^+) & 1/(4R^+) \end{pmatrix}$$

is a basis of Λ^*.

Algorithm 3 ALGPIP-DUAL

Input: Discriminant Δ, integer q such that $2q < \Delta(\ln \Delta)^2 < 4q$.
Output: An approximation of a vector from $8q\Lambda^*$

1. (initial state) $|0\rangle|0\rangle|(1, \Delta \bmod 2)\rangle$.
2. (create superposition) $\longrightarrow \frac{1}{q} \sum_{x_1=0}^{q-1} \sum_{x_2=0}^{q-1} |x_1\rangle|x_2\rangle|(1, \Delta \bmod 2)\rangle$.
3. (compute PIP)
   $\longrightarrow \frac{1}{q} \sum_{x_1=0}^{q-1} \sum_{x_2=0}^{q-1} |x_1\rangle|x_2\rangle| \text{PIP}(x_1, x_2)\rangle$.
4. (measure the third register)

$$\frac{1}{\sqrt{p}} \sum_{x_1=0}^{\lfloor (q-x_1'-1)/n \rfloor} \sum_{x_2 \in \mathcal{M}} \sum_{m=0}^{m_{(\mathbf{x}, x_2')}} |x_1' + x_1 n\rangle|x_2' - x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x}, x_2')}\rangle| \text{PIP}(\mathbf{x}')\rangle,$$

with random $x_1 \in \{0, \ldots, n-1\} \times \{0, \ldots, \lfloor 4R^+ \rfloor\}$, $m_{(\mathbf{x}, x_2')}$ and $\epsilon_{(\mathbf{x}, x_2')}$ as defined in Lemma 4, $\mathcal{M} = \mathcal{M}_{x_1, \mathbf{x}'} = \{x_2 \in \mathbb{Z} \mid 0 \leq x_2' - x_1 S + 4x_2 R^+ + \epsilon_{(\mathbf{x}, x_2')} < q\}$, and $p = \sum_{x_1=0}^{\lfloor (q-x_1-1)/n \rfloor} \sum_{x_2 \in \mathcal{M}} (m_{(\mathbf{x}, x_2')} + 1)$.
5. (apply QFT to the first two registers)

$$\frac{1}{8q\sqrt{p}} \sum_{y_1, y_2=0}^{8q-1} \sum_{x_1=0}^{\lfloor (q-x_1-1)/n \rfloor} \sum_{x_2 \in \mathcal{M}} \sum_{m=0}^{m_{(\mathbf{x}, x_2')}} \exp\left(2\pi i \frac{x_1' + x_1 n}{8q} y_1\right) |y_1\rangle \times$$

$$\times \exp\left(\frac{x_2' - x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x}, x_2')}}{8q} y_2\right) |y_2\rangle| \text{PIP}(\mathbf{x}')\rangle.$$

6. Measure and return the first two registers (y_1, y_2).

Theorem 3. *The set of approximations for vectors from $8q\Lambda^*$ is*

$$\mathcal{Y} = \{ (y_1, y_2) \in \mathbb{Z}^2 \mid 0 \leq y_1 < 8q \text{ and } \frac{y_1}{8q} = \frac{z_1}{n} + \frac{z_2 S}{4nR^+} + \omega_1 \text{ with } z_1, z_2 \in \mathbb{Z} \text{ and } |\omega_1| \leq \frac{1}{16q}$$

$$0 \leq y_2 < \frac{q}{m_{max} + 2} \text{ and } \frac{y_2}{8q} = \frac{z_2}{4R^+} + \omega_2 \text{ with } |\omega_2| \leq \frac{1}{16q} \}.$$

(4)

ALGPIP-DUAL computes vectors $(y_1, y_2) \in \mathcal{Y}$ in quantum polynomial time with probability at least 2^{-16} and requires at most $3 \log(\Delta) + 4 \log \ln \Delta + N$ qubits, where N is the number of temporary qubits which are necessary to execute operations on forms to compute Reg.[7]

---

[6] By $\mathbf{x}^t$, we denote the transpose of the vector $\mathbf{x}$
[7] In [Sch07], it it shown that $N < 10.5 \log \Delta + O(\log^2(\log \Delta))$

Proof. The probability to measure a $\mathbf{y} \in \mathcal{Y}$ is

$$\Pr(\mathbf{y} \in \mathcal{Y}) = \frac{1}{64q^2p} \left| \sum_{x_1=0}^{\lfloor (q-x_1-1)/n \rfloor} \sum_{x_2 \in \mathcal{M}} \sum_{m=0}^{m_{(\mathbf{x},x_2')}} e^{\frac{2\pi i}{8q}\left(y_1 x_1 n + (-x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x},x_2')})y_2\right)} \right|^2. \tag{5}$$

We have

$$x_1 n \frac{y_2}{8q} + (-x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x},x_2')})\frac{y_2}{8q} =$$

$$x_1 n \left(\frac{z_1}{n} + \frac{z_2 S}{4nR^+} + \omega_1\right) + (-x_1 S + 4x_2 R^+ + m + \epsilon_{(\mathbf{x},x_2')})\left(\frac{z_2}{4R^+} + \omega_2\right) \equiv$$

$$x_1 n \omega_1 + (-x_1 S + 4x_2 R^+)\omega_2 + (m + \epsilon_{(\mathbf{x},x_2')})\frac{y_2}{8q} \pmod{1},$$

where $|\omega_1|, |\omega_2| \le 1/(16q)$ and $0 \le y_2 < q/(m_{max} + 2)$. Hence, the sum in (5) is a sum of p vectors of length one which all lie in a segment of size $\pi/2$. This implies $\Pr(\mathbf{y} \in \mathcal{Y}) = |p\sqrt{2}/2|^2/(64q^2p) \ge p/(128q^2)$.

Next, we estimate the lower bound for p and card $\mathcal{Y}$. We have

$$p = (\lfloor (q - x_1 - 1)/n \rfloor + 1) \sum_{x_2 \in \mathcal{M}} (m_{(\mathbf{x},x_2')} + 1) \ge \frac{q}{n}\frac{q}{8R^+}(m_{min} + 1) \text{ and}$$

$$\text{card } \mathcal{Y} = \text{card}\left\{ (z_1, z_2) \in \mathbb{Z}^2 \mid 0 \le \frac{z_1}{n} + \frac{z_2 S}{nR^+} + \omega_1 < 1 \text{ and} \right.$$

$$\left. 0 \le \frac{z_2}{4R^+} + \omega_2 \le \frac{1}{8(m_{max}+2)}, \text{ with } |\omega_1|, |\omega_2| \le \frac{1}{16q} \right\}$$

$$\ge \text{card}\left\{ (z_1, z_2) \in \mathbb{Z}^2 \mid \frac{n}{16q} \le z_1 < n\frac{16q-1}{16q} \text{ and } 1 \le z_2 \le \frac{R^+}{2(m_{max}+2)} - 1 \right\}$$

$$\ge \frac{nR^+}{8(m_{max}+2)}.$$

From the above results, it follows

$$\sum_{y \in \mathcal{Y}} \Pr(y \in \mathcal{Y}) \ge \frac{nR^+}{8(m_{max}+2)}\frac{q}{n}\frac{q}{8R^+}(m_{min}+1)\frac{1}{128q^2} \ge \frac{1}{2^{16}}.$$

The number of qubits can be determined as follows. Each of the first two registers requires at most $\log \Delta + 2 \log(\ln \Delta)$ qubits to keep $q < (1/2)\Delta(\ln \Delta)^2$. As in algorithm REGULATOR-DUAL, the third register requires $\log \Delta + 2$ qubits. □

Algorithm 4 ALGPIP

Input: Reduced form g of discriminant Δ, regulator R^+.
Output: "fail", "not principal", or $\delta(g)$, if g is principal

1. If $R^+ < 64 \ln \Delta$, classically compute and return the solution.
2. Use SAMPLEDUAL-RQ to compute (y_1, y_2) and (y_1', y_2')
3. Set $z_2 = \lfloor y_2 R^+/(2q) \rceil$ and $z_2' = \lfloor y_2' R^+/(2q) \rceil$ and compute $k_1, k_2 \in \mathbb{Z}$ such that $k_1 z_2 + k_2 z_2' = \gcd(z_2, z_2')$.
4. If $\gcd(z_2, z_2') = 1$, then set $p = y_1 k_1 + y_1' k_2 \mod 8q$ and $S' = pR^+/8q$. In this case S is an approximation for S. If $\gcd(z_2, z_2') > 1$, return "fail".
5. Test whether S' is an approximation for S. If not, return "not principal".
6. Return the approximation S' (improve it classically, if necessary).

Theorem 4. ALGPIP *solves the principal ideal problem in a real-quadratic number field* $\mathbb{Q}(\sqrt{\Delta})$ *for every reduced form g in quantum-polynomial time* $O(\text{polylog}(\log \Delta))$. *It is a Monte Carlo type algorithm with success probability at least* 2^{-37}. *The algorithm requires at most* $3\log(\Delta) + 2\log\ln\Delta + N$ *qubits, where N is the number of temporary qubits which are necessary to execute operations on forms to compute* Reg.

Proof. We use the same notation as in the theorem and the algorithms.

First, we test classically whether $R^+ < 64\ln\Delta$ and, if so, the problem can be solved in classical polynomial time using algorithms from [BB94] or [Mau00].

Now, we assume that $R^+ \geq 64\ln\Delta$. With probability at least $2^{-3}2^{-32}$, the quantum subroutine ALGPIP-DUAL returns two different vectors $(y_1, y_2), (y_1', y_2') \in \mathcal{Y}\backslash\{0, 0\}$. By (4)

$$\frac{y_2}{8q} = \frac{z_2}{4R^+} + \omega_2, \quad |\omega_2| \leq \frac{1}{16q},$$

which implies

$$z_2 = \frac{y_2 R^+}{2q} - 4R^+\omega_2 = \frac{y_2 R^+}{2q} + \omega' = \left\lfloor \frac{y_2 R^+}{2q} \right\rceil, \quad |\omega'| \leq \frac{1}{4}.$$

Analogically, $z_2' = \lfloor y_2' R^+/(2q) \rceil$. Using an extended GCD algorithm, we compute $k_1, k_2 \in \mathbb{Z}$ such that $k_1 z_2 + k_2 z_2' = \gcd(z_2, z_2')$. We assume $\gcd(z_2, z_2') = 1$ which is true with probability at least $6/\pi^2$.

Next, assume g is a principal form. In this case $n = 1$. Using (4), we can write

$$\frac{y_1 k_1 + y_1' k_2}{8q} = k_1 z_1 + k_2 z_1' + \frac{S}{R^+} + \omega, \quad |\omega| \leq \frac{k_1 + k_2}{16q} < \frac{1}{R^+}.$$

From $k_1 z_1 + k_2 z_1' \in \mathbb{Z}$ and $0 \leq S/R^+ < 1$, it follows that $S' = pR^+/(8q)$, $p = y_1 k_1 + y_1' k_2 \mod 8q$, is an approximation of S. Now, we test classically whether this is true and, if so, we improve the approximation classically with algorithms from [Mau00]. If S' is not an approximation for S, then our assumption is wrong and g is not a principal form.

Finally, we estimate the success probability of ALGPIP which is the probability to measure two different non-zero vectors from $\mathcal{Y}$ such that $\gcd(z_2, g_2') = 1$. This probability is at least $2^{-35}6/\pi^2 > 2^{-36}$.

The number of qubits follows directly from Theorem 3. $\qquad\square$

Notice, if the output of ALGPIP is "not principal", then we cannot decide whether it is correct or not. However, this case can be solved by applying more advanced techniques from [BP89] and [BK93] for finding a basis of a lattice given approximations for vectors from the dual lattice.

However, if the output of ALGPIP is a distance δ, we can easily test classically whether this distance is correct. This case is sufficient to break the cryptosystem proposed in [BW90], since in this cryptosystem, g is always principal by construction.

6 Conclusion

In this paper, we presented polynomial-time quantum algorithms for solving the regulator and the principal ideal problem in real-quadratic number fields by using functions which are many-to-one on a period. These algorithms reduce the number of qubits by at least $2\log\Delta$ compared to Hallgren's algorithms. This is due to the facts that the period of the lattice is smaller ($8R$ vs. $\lceil\sqrt{\Delta}\rceil R$), the necessary precision for natural algorithms is smaller ($1/8$ vs. $1/\sqrt{\Delta}$), and the function value of Reg and PIP is a form and not a pair of a form and a distance.

An open problem is whether this method can be used for computing the class group of a real-quadratic number field and for improving the algorithms for number fields of degree greater than two which are presented in [Hal05] and [SV05].

References

BB94. I. Biehl and J. Buchmann. Algorithms for quadratic orders. In *Proceedings of Symposia in Applied Math.*, volume 48, pages 425–449, 1994.

BK93. J. Buchmann and V. Kessler. Computing a reduced lattice basis from a generating system. Technical Report ??/93, Technische Universität Darmstadt, Fachbereich Informatik, 1993.

BL95. D. Boneh and R. Lipton. Quantum cryptanalysis of hidden linear functions (extended abstract). In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*. Springer, 1995.

BP89. J. Buchmann and M. Pohst. Computing a lattice basis from a system of generating vectors. In James H. Davenport, editor, *EUROCAL 1987*, volume 378 of *Lecture Notes in Computer Science*, pages 54–63. Springer, 1989.

BV07. Johannes Buchmann and Ulrich Vollmer. *Algorithms for binary quadratic forms*. Springer, 2007.

BW90. Johannes Buchmann and Hugh C. Williams. A key-exchange system based on real quadratic fields. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 335–343. Springer-Verlag, 1990.

Hal02a. Lisa Hales. *The Quantum Fourier Transform and Extensions of the Abelian Hidden Subgroup Problem*. PhD thesis, UC Berkeley, 2002.

Hal02b. Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. In *STOC*, pages 653–658. ACM Press, 2002.

Hal05. Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *STOC*, pages 468–474, 2005.

Hal07. Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *J. ACM*, 54(1), 2007.

HH00. Lisa Hales and Sean Hallgren. An improved quantum fourier transform algorithm and applications. In *IEEE Symposium on Foundations of Computer Science*, pages 515–525, 2000.

Hua82. Loo-Keng Hua. *Introduction to Number Theory*. Springer, New York, 1982.

JW09. Michael J. Jacobson, Jr. and Hugh C. Williams. *Solving the Pell Equation*. Springer-Verlag, 2009.

Len82. Hendrik W. Lenstra, Jr. On the calculation of regulators and class numbers of quadratic fields. In J. V. Armitage, editor, *Journees Arithmetiques, Exeter 1980*, volume 56 of *London Mathematical Society Lecture Notes Series*, pages 123–150. Cambridge University Press, 1982.

Mau00. Markus Maurer. *Regulator approximation and fundamental unit computation for real quadratic orders*. PhD thesis, Technische Universität Darmstadt, 2000.

ME99. Michele Mosca and Artur Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. *Lecture Notes in Computer Science*, 1509:174–188, 1999.

Sch06. Arthur Schmidt. Quantum algorithm for solving the discrete logarithm problem in the class group of an imaginary quadratic field and security comparison of current cryptosystems at the beginning of quantum computer age. In Günter Müller, editor, *ETRICS*, volume 3995 of *Lecture Notes in Computer Science*, pages 481–493. Springer, 2006.

Sch07. Arthur Schmidt. *Zur Lösung von zahlentheoretischen Problemen mit klassischen und Quantencomputern*. PhD thesis, Technische Universität Darmstadt, 2007.

Sei01. Jean-Pierre Seifert. Using fewer qubits in shor's factorization algorithm via simultaneous diophantine approximation. In David Naccache, editor, *CT-RSA*, volume 2020 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 2001.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

SV05. Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *STOC*, pages 475–480, 2005.